

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

APLICACIÓN MÓVIL DE RECOMENDACIÓN CONTEXTUALIZADA DE MÚSICA

ANA CARMONA ANGULO

Tutor: Ignacio Fernández Tobías

Ponente: Iván Cantador Gutiérrez

JULIO 2014

Universidad Autónoma de Madrid
Escuela Politécnica Superior



**Aplicación móvil de recomendación
contextualizada de música**

Ana Carmona Angulo

Julio – 2014

Resumen

En los últimos años, debido a la expansión del uso de la Web, los usuarios se ven cada vez más desbordados en sus tareas de búsqueda y recuperación de información por la ingente cantidad de datos a los que pueden acceder. Por este motivo, los sistemas de recomendación han tenido un éxito y uso crecientes. Un sistema de recomendación es una técnica de filtrado de información que permite a un usuario acceder a los datos útiles para él atendiendo a sus preferencias (gustos, intereses, objetivos).

Existen numerosos tipos de sistemas de recomendación, y se ha observado la importancia del contexto (día y hora, localización del usuario, estado emocional, compañía, etc.) para mejorar la eficacia de las recomendaciones. Por ejemplo, no es lo mismo recomendar un hotel a una pareja para su luna de miel que para una familia.

El contexto de un usuario se compone de las fuentes de influencia que pueden hacerle variar en sus decisiones. Algunas de estas fuentes pueden ser externas, como por ejemplo, la hora, o internas, como el estado de ánimo. Hoy en día, se pueden obtener muchos factores del contexto de un usuario gracias a los smartphones. Prácticamente todas las personas llevamos un teléfono móvil encima en cada momento.

Por otro lado, la importancia de la música en la vida de las personas es evidente. Además, el consumo de música se ha multiplicado precisamente por el uso del teléfono móvil, ya que es muy fácil acceder a música a través de él.

Por estos motivos, se ha visto la importancia de la unión de los sistemas de recomendación con la música y el contexto de usuario. Así, el objetivo de este proyecto es el diseño, implementación y evaluación de una aplicación móvil que recomiende música contextualizada al usuario.

El objetivo principal perseguido con la aplicación desarrollada es intentar concluir qué tipo de recomendaciones son más eficaces: las sensibles al contexto o las que no lo tienen en cuenta y solo hacen uso de las preferencias del usuario. Para ello, se ha realizado un estudio preliminar con usuarios de la aplicación. Los resultados obtenidos en el experimento muestran que un equilibrio entre preferencias de usuario y contexto es beneficioso para alcanzar mejores recomendaciones.

Palabras clave

Sistemas de recomendación, sistema de recomendación basado en contenido, sistema de recomendación de música, contexto, estado de ánimo.

Abstract

In the last years, due to the expansion of the Web, and the huge amount of information that can be accessed on the Internet, people are overwhelmed in their information search and retrieval tasks. For this reason, recommender systems have been successful and their use has been increased unexpectedly. A recommender system is an information filtering technique that lets a user access to useful and relevant data for him according to her preferences (tastes and interests).

There are many types of recommender systems, but in most of them, it has been shown the importance of considering the user's context (e.g., current time and date, the user's location, mood, and social company) to improve the effectiveness of the personalized recommendations. For instance, it is not the same to recommend a hotel to a couple for their honey moon than to recommend a hotel to a family with children.

The user's context consists in influence sources (signals) that can make him change his current decisions. Some of these sources are external, like the time, while others are internal, like the user's mood. Today, we can obtain many of these context's sources by means of a mobile phone. Almost every person is carrying a mobile phone every time, everywhere.

On the other hand, the importance of music in the people's life is evident. And, thanks to the expansion of the use of mobile phones, music consumption has been increased exponentially, since it is very easy to access and listen to music.

For these reasons, we have seen the importance of the union of recommender systems, music and user's context. Thus, the target of this project is to design, implement and evaluate a mobile application that recommends contextualized music to the user.

The main goal pursued with the developed application is to conclude which types of recommendations are more effective: the ones that are aware of the user's context or the ones that are only based on the user's preferences. To achieve such goal, we have conducted a preliminary study with real users. Among other things, the results obtained in the experiment have shown that a balance between the two options lets achieve the best recommendations.

Key words

Recommender system, context-aware recommender system, music recommender system, context, mood.

APLICACIÓN MÓVIL DE RECOMENDACIÓN CONTEXTUALIZADA DE MÚSICA

Índice de contenidos

1. Introducción	1
1.1 Motivación y definición del problema	1
1.2 Objetivos del proyecto	2
1.3 Estructura del documento	3
2. Trabajo relacionado.....	5
2.1 Sistemas de recomendación	5
2.2 Sistemas de recomendación sensibles al contexto.....	6
2.3 Sistemas de recomendación de música	9
3. Señales contextuales consideradas	11
3.1 Tiempo	11
3.2 Localización	12
3.3 Estado de ánimo.....	12
4. Sistema de recomendación propuesto	15
4.1 Arquitectura global.....	15
4.2 Módulo de modelaje de usuarios e ítems.....	16
4.2.1 Datos del usuario.....	17
4.2.2 Datos del artista.....	17
4.2.3 Géneros musicales	17
4.3 Módulos de adquisición de señales contextuales	18
4.3.1 Sensor de tiempo.....	18
4.3.2 Sensor de localización	19
4.3.3 Sensor de estado de ánimo.....	20
4.4 Módulo de acceso a datos	21
4.4.1 Gestión de usuarios.....	21
4.4.2 Gestión de ítems	22
4.4.3 Gestión de contextos.....	23
4.4.4 Gestión de interacciones	23
4.5 Módulo de recomendaciones	25
4.5.1 Módulo de filtrado basado en contenido.....	26
4.5.2 Módulo de filtrado contextual.....	28
4.5.3 Normalización de puntuaciones	32

5. Aplicación móvil desarrollada	33
5.1 Tecnologías empleadas	33
5.2 Componentes principales de la interfaz gráfica.....	34
5.2.1 Animación de presentación	35
5.2.2 Menú de login	35
5.2.3 About	36
5.2.4 Nuevo Usuario	36
5.2.5 Selección de géneros musicales	37
5.2.6 Menú de usuario	37
5.2.7 Selección de ubicación	38
5.2.8 Selección de estado de ánimo	38
5.2.9 Lista de recomendaciones	39
5.3 Flujo de la aplicación.....	41
5.4 Evaluación preliminar	42
5.4.1 Diseño del experimento.....	42
5.4.2 Metodología de la evaluación	43
5.4.3 Conjunto de datos	45
5.4.4 Resultados obtenidos	50
5.4.5 Discusión	55
6. Conclusiones y trabajo futuro	57
6.1 Conclusiones	57
6.2 Trabajo futuro	58
6.2.1 Modificación del sensor de localización	58
6.2.2 Mejora del perfil de usuario.....	59
6.2.3 Inferencia de gustos musicales.....	60
6.2.4 Mejora del sensor de estados de ánimo.....	61
6.2.5 Recomendación de canciones	64
6.2.6 Anotación de sentimientos para canciones y usuarios.....	64
7. Referencias.....	65

Índice de tablas

Tabla 1 : Puntuación de ítems en sistemas de recomendación	5
Tabla 2 : Localizaciones disponibles	12
Tabla 3 : Estados de ánimo disponibles	13
Tabla 4 : Géneros musicales disponibles	17
Tabla 5 : Tabla de usuarios en base de datos.....	21
Tabla 6 : Tabla de artistas en base de datos.....	22
Tabla 7 : Tabla de géneros en base de datos.....	22
Tabla 8 : Tabla de contextos en base de datos	23
Tabla 9 : Tabla de ratings en base de datos.....	24
Tabla 10 : Tabla de cambios de estado de ánimo en la base de datos.....	24
Tabla 11 : Ejemplo de vectorización de usuarios y artistas	26
Tabla 12 : Ejemplo de contextos en base de datos	29
Tabla 13 : Ejemplo de artistas en la base de datos.....	46
Tabla 14 : Transiciones de estados de ánimos.....	53
Tabla 15 : Ejemplo de diccionario de sentimientos.....	63

Índice de ilustraciones

Ilustración 1 : Comparación de distintos algoritmos de recomendación.....	8
Ilustración 2 : Relación entre afecto, emoción y estados de ánimo	13
Ilustración 3 : Representación de emociones según el modelo de Russell.....	14
Ilustración 4 : Arquitectura de la aplicación	15
Ilustración 5 : Sensor de estados de ánimo	20
Ilustración 6 : Diagrama de transiciones de pantallas de la aplicación.....	34
Ilustración 7 : Pantalla de animación	35
Ilustración 8 : Pantalla del menú de login de la aplicación.....	35
Ilustración 9 : Diálogo de información de la aplicación	36
Ilustración 10 : Pantalla de creación de un nuevo usuario.....	36
Ilustración 11 : Pantalla de selección de géneros musicales	37
Ilustración 12 : Pantalla del menú de usuario de la aplicación.....	37
Ilustración 13 : Pantalla de selección de localización	38
Ilustración 14 : Pantalla de selección de estado de ánimo	38
Ilustración 15 : Pantallas de recomendaciones	39
Ilustración 16 : Diálogo de información del artista	39
Ilustración 17 : Pantalla de puntuación a un artista.....	40
Ilustración 18 : Diálogo de cambio de estado de ánimo	40
Ilustración 19 : Diagrama de flujo de la aplicación	41
Ilustración 20 : Dominio de datos en un sistema de recomendación	43
Ilustración 21 : Conjunto de usuarios	45
Ilustración 22 : Preferencias musicales de los usuarios	46
Ilustración 23 : Géneros musicales de los artistas	47
Ilustración 24 : Contexto temporal de los usuarios	48
Ilustración 25 : Ubicación de los usuarios	48
Ilustración 26 : Estado de ánimo inicial de los usuarios.....	49
Ilustración 27 : Ítems puntuados por λ	50
Ilustración 28 : Precisión @k.....	51
Ilustración 29 : Recall @k.....	51
Ilustración 30 : Media armónica @k.....	52
Ilustración 31 : Transiciones de estados de ánimo	54
Ilustración 32 : Ejemplo de metadatos musicales	60
Ilustración 33 : Lynguo.....	62
Ilustración 34 : Gráfica de Lynguo - Opinión de los usuarios sobre la Selección Española de fútbol durante el día 12 de junio de 2014	62
Ilustración 35 : Opinión de los usuarios sobre la proclamación del rey Felipe VI desde el 18 de junio hasta el 22 de junio de 2014	63

Índice de ejemplos de código

Código 1 : Llamada al objeto recomendador	25
Código 2 : Implementación del algoritmo de recomendación basado en contenido.....	27
Código 3 : Implementación del algoritmo de recomendación basado en contexto.....	30
Código 4 : Similitud de un ítem con un contexto	31
Código 5 : Normalización de puntuaciones.....	32

Introducción

En este capítulo se aportará una breve descripción de la motivación, los objetivos y la estructura del documento del trabajo realizado.

1.1 Motivación y definición del problema

En los últimos años se ha experimentado un crecimiento exponencial en la información a la que podemos acceder. Así, para el usuario corriente, es prácticamente imposible manejar tantos datos de manera autónoma. Este problema se pone de manifiesto cuando necesitamos realizar una búsqueda en Internet. Por este motivo se crean los **sistemas de recomendación**.

Un sistema de recomendación es un tipo de técnica de filtrado de información que permite al usuario recibir aquellos datos que le sean de utilidad. El objetivo de estos sistemas es hallar ítems que sean de mayor relevancia al usuario teniendo en cuenta sus preferencias y sus necesidades y, a diferencia de los sistemas de búsqueda tradicionales, sin requerir una consulta explícita. Estos sistemas se han aplicado en numerosos dominios: películas, música, libros, restaurantes, etc.

En numerosos casos, se ha observado que es beneficiosa la obtención y consideración de **información contextual** del usuario para poder ofrecerle mejores recomendaciones. Por ejemplo, para recomendar una película con restricción de edad, es importante saber la edad del usuario. Sin embargo, la aplicación de procedimientos para obtener el contexto de usuario suele ser muy costosa. Por este motivo, se han meditado diversas opciones para hacer este proceso más eficiente.

Una fuente importante de información contextual de un usuario, es el **teléfono móvil** [7]. En la actualidad, prácticamente todas las personas llevamos un teléfono encima durante todo el día. Gracias a los datos que se pueden obtener de estos aparatos, recoger información contextual se hace una tarea más sencilla. Otro beneficio que nos ofrece un teléfono móvil es la posibilidad de escuchar música en cualquier lugar, ya sea almacenada en el propio dispositivo o accediendo a ella a través de la red.

Actualmente, hay muchos sistemas de recomendación de música y muchos sistemas de recomendación sensibles al contexto, pero pocos son los sistemas que combinen las dos opciones: recomendación de música sensible al contexto del usuario.

Al observar esta carencia en los sistemas de recomendación, se planteó la implementación de un sistema de recomendación que combinara las dos tareas: **recomendar música teniendo en cuenta el contexto del usuario**. Por los beneficios que aportan los teléfonos móviles, se decidió que la mejor manera de desarrollar este sistema era en una aplicación móvil.

La motivación de la creación de este sistema es la necesidad del usuario de un filtro de información para obtener los datos que realmente le interesan. Y dentro de esta necesidad, es importante destacar la contextualización, ya que las necesidades de los usuarios varían dependiendo del contexto en el que se encuentren.

Para hacer más sencilla la contextualización, se ha optado por una aplicación móvil porque, gracias a la telefonía móvil, se dispone de una cantidad ingente de información contextual del usuario (dónde se encuentra, qué fecha es, qué hora...). Y se ha decidido recomendar música porque es un tipo de ocio altamente consumible por los usuarios y, además, es corto en duración: una canción dura aproximadamente dos minutos, mientras que una película dura dos horas.

1.2 Objetivos del proyecto

El objetivo de este proyecto es implementar una aplicación móvil de recomendación de música que sea consciente del contexto del usuario.

Para poder cumplir este objetivo, es necesario poder representar y capturar la información contextual de un usuario y obtener sus preferencias musicales. También es necesaria la implementación de un algoritmo capaz de sugerir música al usuario en base a sus gustos y al contexto en el que se encuentre.

Otro objetivo de proyecto, aunque no el propósito principal, es la evaluación de la aplicación desarrollada, es decir, comprobar si los usuarios encuentran útiles las recomendaciones, y, en particular, confirmar si se consideran más apropiadas las recomendaciones sensibles al contexto, las que no lo tienen en cuenta, o las que ponderan los dos factores.

Otra conclusión que se desea obtener es la localización de patrones de transiciones de estados de ánimo (considerado como señal contextual), es decir, si hay más probabilidades de, comenzando en un determinado estado de ánimo, permanecer en este o si, por el contrario, es más frecuente cambiar a otro, dadas unas recomendaciones de música u otras.

1.3 Estructura del documento

Este documento se distribuye en siete capítulos, cada uno de ellos informa sobre una parte importante del diseño, implementación o evaluación de la aplicación desarrollada.

- En el **primer capítulo** se presenta una introducción al trabajo realizado, junto con la motivación del mismo y los objetivos que se persiguen.
- En el **segundo capítulo** se revisa trabajo relacionado en el área de los sistemas de recomendación. En concreto, se describen los sistemas de recomendación sensibles al contexto y los sistemas de recomendación de música.
- En el **tercer capítulo** se describen y se motivan las señales contextuales consideradas para la aplicación desarrollada en este trabajo.
- El **capítulo cuarto** está dedicado a la arquitectura de la aplicación desarrollada, se comentan los módulos que la forman y se explican los procedimientos llevados a cabo para su implementación.
- En el **quinto capítulo** se presenta la aplicación móvil implementada. Los componentes de la interfaz gráfica, que es la que se encarga de gestionar las interacciones con los usuarios, se describen en este punto. Además, se incluye la evaluación preliminar de la aplicación.
- En el **sexto capítulo** se detallan las conclusiones del trabajo y se presentan varias líneas de posible trabajo futuro a desarrollar.

2

Trabajo relacionado

En este apartado se explica qué es un sistema de recomendación y qué tipos hay. Se especifican las diferencias entre ellos y los componentes de cada uno. Se da especial importancia a los sistemas de recomendación de música.

2.1 Sistemas de recomendación

El objetivo de los **sistemas de recomendación** es encontrar aquellos ítems (canciones, películas, noticias, restaurantes, etc.) que sean de mayor utilidad para el usuario, teniendo en cuenta sus gustos y preferencias, sin que esto requiera que realice una consulta explícita.

En muchos sistemas de recomendación los usuarios expresan sus preferencias sobre los ítems mediante evaluaciones numéricas o *ratings*, típicamente en una escala de 1-5 o de 1-10. Estas interacciones se suelen formalizar a través de una cierta *función de utilidad*

$$G: \text{Usuario} \times \text{Ítem} \rightarrow \text{Rating}$$

que codifica cómo de relevante es un determinado ítem para un usuario. Puesto que en aplicaciones reales los usuarios no evalúan todos los ítems del catálogo, esta función no está definida para todos los pares (usuario, ítem).

En la literatura es común representar la función de utilidad en forma de matriz, por ejemplo:

Ítem	User	u1	u2	u3	u4	u5
i1		5			9	2
i2		3		2	8	4
i3			1		6	4
i4		8	7			6
i5		9	4	6	8	

Tabla 1 : Puntuación de ítems en sistemas de recomendación

En general la matriz no está completa. El objetivo de un sistema de recomendación es estimar las preferencias de un usuario acerca de objetos no conocidos y sugerirle aquellos para los que predice una mayor utilidad.

Los sistemas de recomendación se suelen clasificar en función de las técnicas que utilizan para calcular las predicciones. Las técnicas más comunes para la obtención de la lista de recomendaciones son [2]:

- **Recomendación basada en contenido** [6]: estos métodos están basados en la descripción del ítem y las preferencias de un usuario. En un sistema de recomendación basado en contenido, se recomiendan elementos parecidos a los que le gustan al usuario.
- **Filtrado colaborativo** [5]: estos métodos están basados en la recolección y análisis de datos en un conjunto de usuarios para obtener las preferencias de los mismos y predecir lo que le gustará a uno de ellos en función de su similitud con otros usuarios.

En este trabajo utilizaremos un método de **recomendación basada en contenido**. En nuestro caso, las preferencias del usuario serán definidas por una lista de géneros musicales, que se compararán con los géneros definidos previamente para los músicos.

2.2 Sistemas de recomendación sensibles al contexto

Los sistemas de recomendación se han convertido en una herramienta extremadamente útil a la hora de predecir el comportamiento de un usuario o aceptación de un producto.

Sin embargo, en muchas ocasiones es necesario utilizar otras variantes para llevar a cabo una correcta recomendación. Por ejemplo, no es lo mismo recomendar paquetes vacacionales para recién casados, que para parejas con niños o para un grupo de adolescentes. También hay que tener en cuenta, en el caso de las vacaciones, la época del año, ya que las vacaciones invernales difieren mucho de las estivales.

Estos hallazgos han cambiado el concepto de marketing: ahora mismo no se trata simplemente de vender una marca, sino de convencer al cliente de que el producto que más se ajusta a sus necesidades (definidas por un contexto) es, efectivamente, el que se le está ofreciendo.

El **contexto** (o información contextual) es un concepto multifacético que se compone de las condiciones y circunstancias (fiablemente comprobadas) que afectan a un hecho o evento concreto.

Por ejemplo, no es lo mismo recomendar una película para una pareja, que para una familia con niños. En este caso, una característica útil del contexto sería la compañía. Actualmente, también existen sistemas de recomendación de restaurantes; en estos sistemas, sería conveniente saber las preferencias gastronómicas de un usuario. Si no le gusta la comida china, lo ideal sería que el usuario lo indicara, y que a partir de ese momento no se le recomendaran ese tipo de restaurantes. O, si el usuario prefiere la comida italiana, se podría dar mayor relevancia a estos restaurantes, pero sin descartar los demás.

Un ejemplo de sistema de recomendación de restaurantes es la web “El tenedor”¹. En esta web, el usuario introduce la zona que más le interesa para ir a comer o cenar y la web se encarga de mostrar los restaurantes que más se ajustan a las preferencias del usuario. En esta web también se tiene en cuenta las búsquedas anteriores del usuario, por lo que, en base a las búsquedas, se va creando un perfil de usuario donde se guardan sus preferencias.

A la hora de predecir la utilidad de un ítem para un usuario, los sistemas de recomendación sensibles al contexto consideran uno o varios componentes adicionales que capturan las circunstancias en las que se produce cada interacción. La *función de utilidad* es ahora:

$$R: \text{Usuario} \times \text{Ítem} \times \text{Contexto} \rightarrow \text{Rating}$$

Contexto define la información contextual relacionada con la aplicación. Hay que recordar que el contexto es un componente formado de otros componentes, como pueden ser la hora o la compañía social. Cada aplicación define el contexto que más le conviene para la correcta recomendación.

Un problema que se presenta ante las recomendaciones sensibles al contexto es la adquisición de la información contextual, ya que puede llegar a ser un proceso muy costoso. La información contextual se puede obtener de numerosas formas, pero las más comunes son:

- **Explícitamente:** de manera directa preguntando a las personas relacionadas u obteniendo esa información de otras fuentes.
- **Implícitamente:** a partir de datos o del ambiente. Por ejemplo, obteniendo la localización del usuario o la marca de tiempo de una transacción.
- **Inferida:** se infiere el contexto usando minería de datos o estadísticas. Para obtener información inferida es necesario construir un modelo y entrenarlo para obtener los datos adecuadamente.

Una vez obtenida la información contextual es necesario aplicarla en el sistema de recomendación para obtener los ratings correspondientes. Es posible aplicar esta información en diferentes etapas. Dependiendo de la etapa en la que se aplique, se compondrá un tipo de sistema de recomendación u otro [1].

- **Pre-filtrado contextual:** la información contextual se aplica en la primera etapa, para obtener un conjunto de datos acorde a ella. Después, los ratings se predicen mediante un sistema de recomendación bidimensional (2D).
- **Post-filtrado contextual:** la información contextual se ignora en un primer momento y los ratings se predicen con un sistema bidimensional. Después de tener todas las predicciones, estas se contextualizan para cada usuario.
- **Modelado contextual:** en este modelo, se usa la información contextual directamente para estimar los ratings.

¹ El tenedor, <http://www.eltenedor.es> - 5/VII/2014

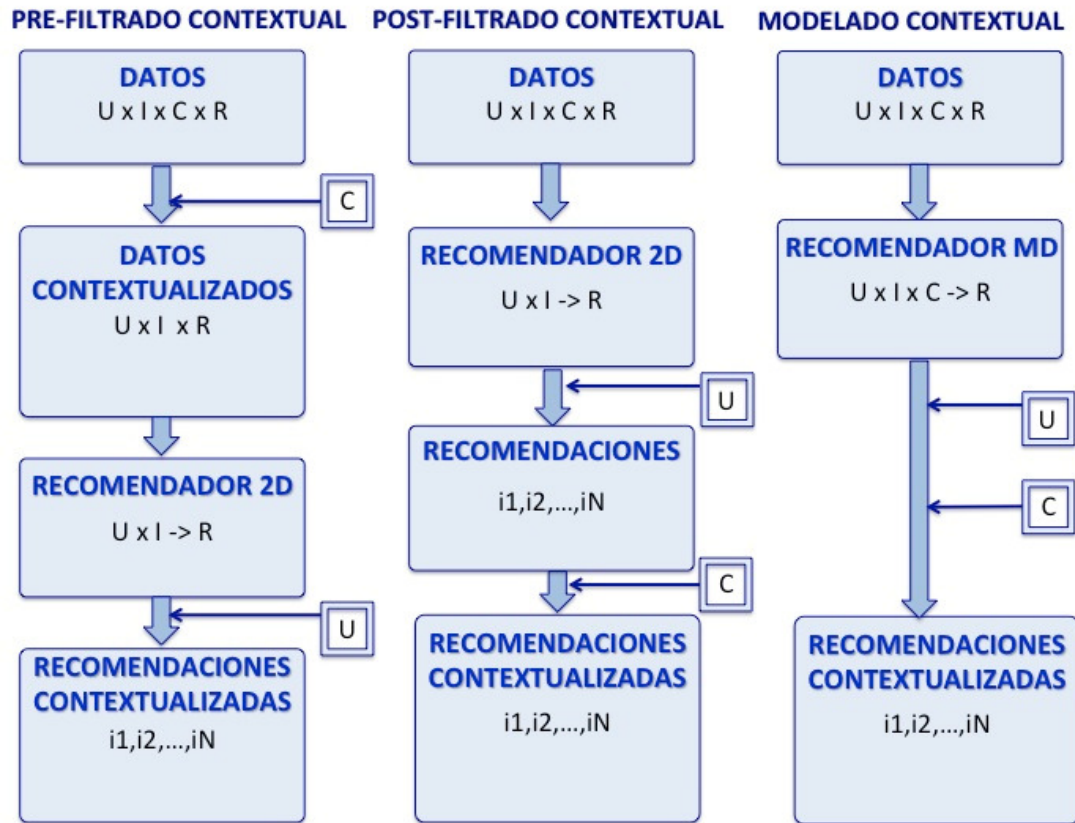


Ilustración 1 : Comparación de distintos algoritmos de recomendación

En muchos casos, es necesario **generalizar** a la hora de definir el contexto. Por ejemplo, ampliar los días de la semana a períodos (días laborables o fin de semana). Esto se hace para que exista mayor similitud entre contextos, es decir, que más de un ítem esté en un contexto.

La generalización del contexto se hace para que coincidan más ítems dentro de un mismo contexto. Definiendo períodos dentro de los componentes de dicho contexto, se crea una mayor afluencia de eventos en cada período y así se pueden agrupar diversos ítems fácilmente.

En nuestro caso, el contexto estará formado por **localización, tiempo y estado de ánimo**. Estas características del contexto, estarán formadas a su vez por otras subcaracterísticas que definiremos en secciones posteriores.

La obtención de estos componentes del contexto se realizará mediante la implementación de sensores. Cada sensor implementa un método de obtención: la localización y el estado de ánimo se obtendrán de una manera explícita y el tiempo de una forma implícita.

En este trabajo, el sistema de recomendación implementado será basado en post-filtrado contextual. Se obtendrá la puntuación de los ítems con los algoritmos de recomendación basados en contenido y, a continuación, se reorganizarán en función de la puntuación obtenida mediante la información contextual.

2.3 Sistemas de recomendación de música

“La **música** es, según la definición tradicional del término, el arte de organizar sensible y lógicamente una combinación coherente de sonidos y silencios utilizando los principios fundamentales de la melodía, la armonía y el ritmo, mediante la intervención de complejos procesos psicoanímicos”².

Actualmente existen numerosos sistemas de recomendación de música. Esto se debe al gran consumo de música que existe entre los usuarios. Hoy en día no existe un perfil definido para el usuario de música. Gracias a los numerosos géneros y variantes musicales, se da cabida a un amplio número de perfiles de usuario, desde la abuelita de 80 años enamorada del pasodoble hasta los niños más pequeños con sus canciones infantiles.

Un ejemplo de sistema de recomendación musical es el de la web “Last.fm”³. En este sistema, se guardan las búsquedas anteriores del usuario para usarlas en una futura recomendación. De esta manera, te recomiendan artistas parecidos a otros que ya has escuchado.

Hoy en día, gracias al extendido uso del teléfono móvil, la música está al alcance de cualquiera, ya sea accediendo a una web que la ofrezca, o simplemente llevando archivos musicales en el aparato.

Esto es extremadamente beneficioso para los sistemas de recomendación conscientes al contexto. El teléfono móvil ofrece un amplio abanico de información del usuario al que el sistema de recomendación puede acceder para perfeccionar sus algoritmos. Por ejemplo, se puede crear un perfil temporal gracias a la hora de las transacciones del teléfono. También es posible crear un perfil de localización para saber dónde se encuentra el usuario y qué música prefiere en ese sitio. Un ejemplo de uso de un perfil de localización sería adecuarse a la música que el usuario prefiere en el trabajo, en el gimnasio o en casa.

La música es, además, un ejemplo de producto reincidente. Esto significa que se tiende a escuchar más veces una misma canción que, por ejemplo, a ver de nuevo una misma película. Es mucho más fácil para un usuario encontrarse con una canción que haya escuchado ya y, aun así, volver a oírla.

Cuando hablamos de recomendación de música, normalmente nos referimos a recomendación de canciones, álbumes o artistas, pero el caso más común es el de recomendación de canciones [9]. En este caso, el posible impacto del error cometido por el sistema de recomendación es ínfimo en comparación con otros sistemas de recomendación.

² Música, Wikipedia, <http://es.wikipedia.org/wiki/Música> - 5/VII/2014

³ Last.fm, www.lastfm.es - 5/VII/2014

Por ejemplo, si le recomendamos una canción a un usuario y resulta que esta no le gusta, únicamente ha “malgastado” tres minutos de su tiempo, sin embargo, con una película el tiempo desaprovechado se puede incrementar hasta las dos horas. Si se recomienda erróneamente un restaurante, el usuario puede perder tiempo y también dinero. El caso más extremo sería el de la recomendación de paquetes vacacionales: si al usuario no le gusta el hotel que se le había recomendado, además de perder una alta suma de dinero, también pierde el tiempo de las vacaciones, que puede llegar a ser de 15 días.

La accesibilidad, el gran número de usuarios y el bajo impacto del error, hacen de la recomendación de música una de las más implementadas.

3

Señales contextuales consideradas

En esta sección se describe la información contextual que se ha considerado para el sistema de recomendación de música desarrollado en este trabajo. En concreto, se detalla la información recogida para cada uno de los tres tipos de señales estudiados: señales temporales, de localización y del estado de ánimo del usuario.

3.1 Tiempo

El momento perfecto de un usuario para escuchar música puede ser completamente diferente al de otro. Este momento también influirá en el tipo de música que elija el usuario.

Por ejemplo, no escuchamos la misma música al levantarnos (cuando la preferencia habitual es no llevarse sobresaltos) que a media mañana (cuando, por ejemplo, podemos estar realizando las tareas de la casa y lo que nos interesa es estar activos).

En este caso, tenemos en cuenta diversos parámetros para definir el contexto temporal en el que se encuentra el usuario.

Se obtiene la marca de tiempo actual, es decir, la hora, con la cual se obtendrán los parámetros necesarios para categorizar el tiempo.

Las categorías que se utilizarán en la aplicación serán:

- **Período del día:** en nuestro caso, se utilizarán los períodos definidos por “mañana”, “tarde” y “noche”.
- **Período de la semana:** se diferenciará entre día laborable y día festivo.
- **Estación del año:** primavera, verano, otoño e invierno.
- **Año:** se considerará el año actual.

3.2 Localización

En este caso tomaremos la localización del usuario de una manera muy sencilla: se le preguntará dónde se encuentra y se le darán diversas opciones como posibles respuestas.

En el trabajo	En casa
En el coche / bus	En un bar / discoteca

Tabla 2 : Localizaciones disponibles

La localización influye en un usuario a la hora de elegir música. Es distinta la música que utilizamos para estudiar (cuando nos interesa estar relajados y concentrados) que la que escuchamos al ir a un bar (cuando queremos estar activos).

Es importante definir una lista limitada de opciones de localización para que sea más fácil comparar la similitud de los contextos. En esta aplicación es necesaria la comparación de contextos porque es una parte del algoritmo de recomendación. Si únicamente damos al usuario cuatro alternativas de ubicación, las probabilidades de que el usuario elija un emplazamiento que ya ha sido insertado en la base de datos aumentan.

Es posible hacer un modelado de la ubicación del usuario mucho más preciso, por ejemplo, añadiendo barrio, ciudad y país, pero para este modelado habría que tener muchos más datos del usuario y, por consiguiente, el tratamiento de estos datos sería más complejo.

3.3 Estado de ánimo

Para poder comprender el resto del documento, es importante definir tres conceptos básicos: afecto, emoción y estado de ánimo.

- **Afecto.** Según la RAE, el afecto es *“cada una de las pasiones del ánimo, como la ira, el amor, el odio, etc., y especialmente el amor o el cariño”*. Es, por tanto, un término genérico que manejaremos como un gran rango de sentimientos. Se trata de un concepto que engloba tanto estados de ánimo como emociones.
- **Emoción.** Una emoción es un sentimiento intenso que está dirigido a alguien o algo.
- **Estado de ánimo.** Se trata de un sentimiento que tiende a ser menos intenso que las emociones y que, normalmente, está asociado a un estímulo contextual.

Las emociones son más transitorias que los estados de ánimo, es decir, se experimentan más emociones que estados de ánimo, ya que estos son más duraderos. Las emociones pueden convertirse en estados de ánimo cuando se olvida el motivo por el cual nació la emoción. En la dirección contraria, los estados de ánimo pueden generar emociones a los estímulos contextuales [3]. Sin embargo, las emociones y los sentimientos se pueden englobar bajo el término *afecto*.



Ilustración 2 : Relación entre afecto, emoción y estados de ánimo

Afecto, emociones y estados de ánimo son distinguibles solo en teoría, en la práctica la división no es siempre tan transparente. Por este motivo, a lo largo de este trabajo se usarán los términos “emoción” y “estado de ánimo” sin distinciones.

En este trabajo se ha optado por utilizar una lista de 16 estados de ánimo definidos según el modelo de afecto de Russell [8], un modelo bien conocido en el ámbito de la psicología.

Tenso	Preocupado
Estresado	Nervioso
Triste	Deprimido
Adormecido	Fatigado
Alerta	Exaltado
Excitado	Feliz
Contento	Sereno
Relajado	Calmado

Tabla 3 : Estados de ánimo disponibles

En el modelo de Russell, el afecto se describe como resultado de dos variables: **valencia** y **excitación**. Las emociones se representan como puntos en un sistema de coordenadas.

La valencia se puntúa en el eje horizontal y la excitación en el vertical.

Cuanto más alta esté una emoción, más excitación conlleva. Cuanto más a la derecha esté la emoción, más valencia.

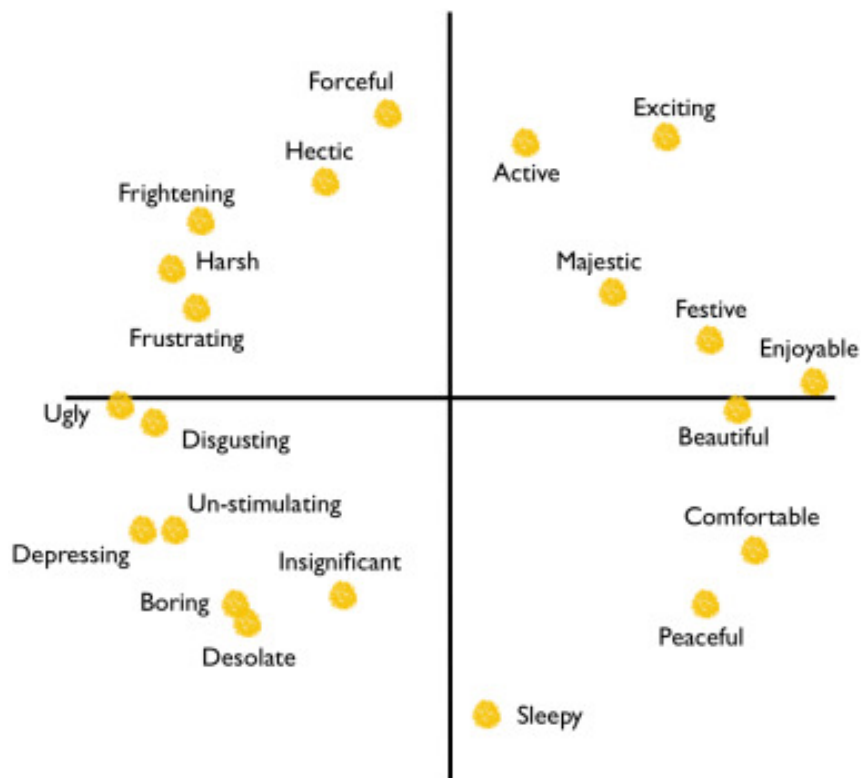


Ilustración 3 : Representación de emociones según el modelo de Russell

4

Sistema de recomendación propuesto

En esta sección se describe la arquitectura de la aplicación móvil que se ha implementado. La aplicación se ha desarrollado de manera modular y se compone de varias partes encargadas de distintas funciones. A continuación se detallan los módulos y se explica su propósito dentro de la aplicación.

4.1 Arquitectura global

La aplicación desarrollada está compuesta por módulos que se relacionan entre sí dando lugar a un esquema modular completo y compacto. Gracias a este diseño, es posible actualizar o reemplazar componentes sin que el resto de la aplicación se vea afectada.

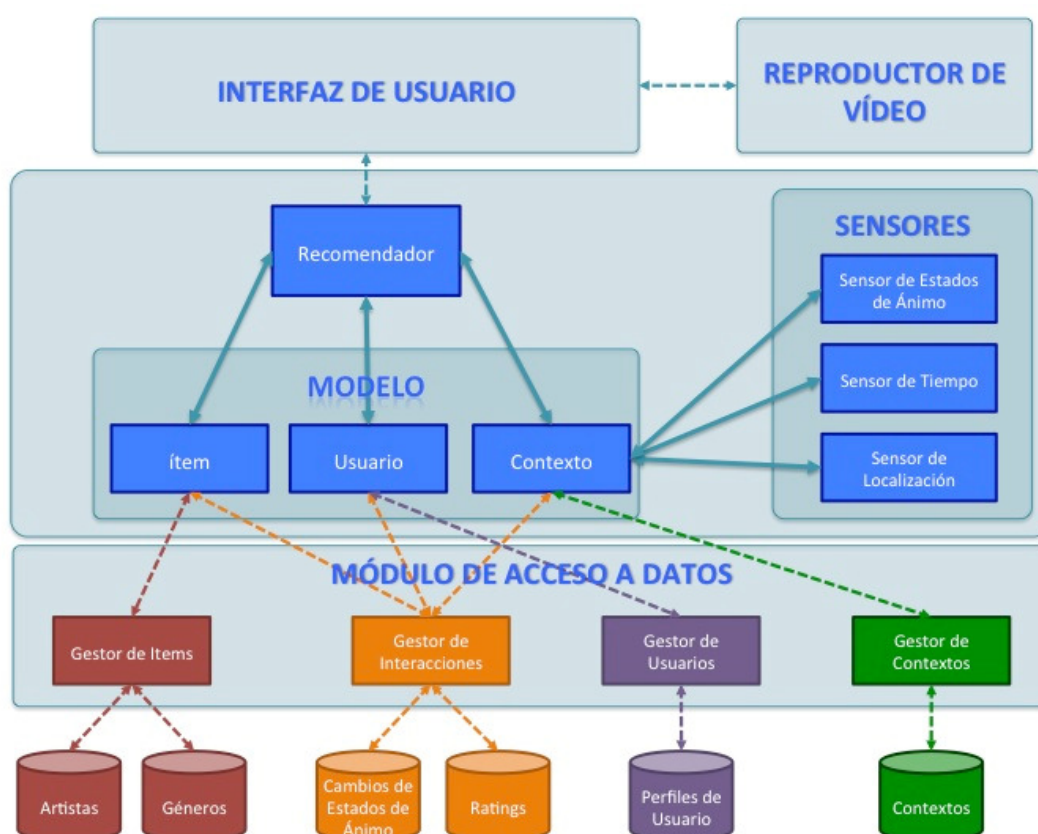


Ilustración 4 : Arquitectura de la aplicación

Como se puede apreciar en la figura anterior, los módulos principales de los que se compone la aplicación son:

- **Módulo de acceso a datos:** se encarga de gestionar las transacciones entre la base de datos y el resto del sistema.
- **Modelo:** el modelo lo forman las clases donde se definen los componentes que representan las distintas entidades que se manejan en la aplicación, esto es, usuarios, ítems y contextos.
- **Sensores:** los sensores se encargan de la obtención de las señales contextuales necesarias para la aplicación, como, por ejemplo, la marca de tiempo.
- **Recomendador:** en este módulo se implementan los algoritmos que calculan las recomendaciones para cada usuario en cada contexto.
- **Interfaz de usuario:** la interfaz de usuario es la encargada de comunicarse con el usuario, esto es, leer los datos de entrada y mostrar los datos de salida.

Estos módulos son agrupaciones de clases que se forman atendiendo a la funcionalidad que proporcionan al sistema. En las siguientes secciones se describen en profundidad los distintos módulos del sistema.

4.2 Módulo de modelaje de usuarios e ítems

En los módulos de modelaje se definen los componentes que representan las distintas entidades de la aplicación.

En esta apartado hablaremos de dos componentes de este módulo: **usuarios** e **ítems**. No se hablará del tercer componente, los contextos, puesto que ya se ha descrito en el capítulo 3.

Se pueden agrupar los usuarios y los ítems (artistas, en este caso) porque se tratan como vectores de géneros musicales.

El objeto “usuario” se compone de tres elementos: **nombre de usuario**, **contraseña** y **géneros musicales preferidos**. El objeto “artista” se compone de cuatro: **nombre del artista**, **información**, **vídeos musicales** y **géneros**.

4.2.1 Datos del usuario

En nuestro caso, los únicos datos que se guardan son el nombre del usuario y la contraseña que utilizará para acceder a la aplicación.

El nombre de usuario debe ser único, ya que con este nombre se le identifica en la aplicación.

Nos hemos limitado únicamente a estos datos de usuario, pero se pueden ampliar para definir perfiles de usuario más sofisticados. Se explicará una posible ampliación de los datos de usuario en la sección “Trabajo futuro”.

4.2.2 Datos del artista

Un artista se compone de un identificador, un nombre, una URL con información del artista, tres vídeos musicales del artista y una lista de géneros musicales.

En este caso, al contrario que en los usuarios, el nombre del artista no tiene que ser único, ya que para conseguir la unicidad de los artistas, se ha añadido un id.

4.2.3 Géneros musicales

Se ofrecen **26 géneros musicales**, de los cuales, un usuario puede seleccionar todos los que quiera. Los géneros musicales son:

Música instrumental	Música ambiental
Blues	Música clásica
Country	Electrónica
Emotive punk	Folk
Hardcore	Hip Hop
Indie	Jazz
Latina	Heavy Metal
Pop	Pop punk
Punk	Reggae
Rhythm and blues	Rock
Soul	Música universal
Música de los años sesenta	Música de los años setenta
Música de los años ochenta	Música de los años noventa

Tabla 4 : Géneros musicales disponibles

Una vez definidos los gustos musicales del usuario (y con el nombre y contraseña fijados), estamos en disposición de guardar el usuario en la base de datos.

Por otro lado, los géneros que se relacionan con un artista son los que este artista interpreta. Como un artista puede relacionarse con más de un género, la relación entre artista y género se pondera mediante unas proporciones. Por ejemplo, si un artista toca rock y jazz, las ponderaciones de ambos géneros en relación con el artista podrían ser 0.7 y 0.3. Estas ponderaciones siempre sumarán 1.

Existe la opción de obtener los gustos musicales de un usuario a través de los metadatos de las canciones que hay en su teléfono móvil o los de las transacciones de las páginas web de música. Así obtendríamos los géneros preferidos del usuario de una manera inferida y transparente para este. De este modo, se conseguiría mucha más información de los gustos reales del usuario, aunque el proceso sería más costoso. Por ello, hemos optado por una versión más asequible, donde se le dan los géneros prefijados al usuario. Se hablará más de esta opción en el apartado “Trabajo futuro”.

4.3 Módulos de adquisición de señales contextuales

Las señales contextuales consideradas son **tiempo**, **localización** y **estado de ánimo**. Se adquirirán mediante unos módulos llamados **sensores**. Gracias a los datos obtenidos por los sensores, se creará un contexto de usuario.

4.3.1 Sensor de tiempo

El sensor de tiempo se encarga de proporcionar la marca de tiempo en el momento de la transacción de datos. Así, definiremos todos los campos del contexto basados en el tiempo.

El contexto utilizado en la aplicación es un **contexto normalizado**, es decir, se han agrupado campos para la correcta categorización de los componentes del sensor de tiempo.

En el momento de la transacción (insertar un rating, por ejemplo), se obtiene la marca de tiempo y, gracias a ella, se obtienen los componentes del contexto de tiempo.

Los componentes del contexto temporal son:

- **Año**: se considera el año actual. En este momento no será de mucha utilidad, pero servirá para que en el futuro, cuando se desarrolle una base de datos más amplia, se pueda recomendar música de distintos años.
- **Período del día**: se define el período del día como “**mañana**”, “**tarde**” y “**noche**”. El momento del día en el que nos encontremos influye en el estado de ánimo de las personas, por este motivo, es muy importante definir bien los períodos del día.

Se considera “mañana” a las horas que van desde las 4:00 hasta las 11:59; “tarde” a las horas que van entre las 12:00 y las 19:59; y consideramos “noche” a las horas restantes, desde las 20:00 hasta las 3:59.

- **Período de la semana:** se definen **días laborables** y **días de fin de semana**. El estado de ánimo también varía dependiendo del día: usualmente, cuando se está de peor humor es los lunes y cuando se está de mejor humor, suele ser los sábados [3]. Como el estado de ánimo influye en el tipo de música que nos apetece escuchar, es importante tener en cuenta las fases en las que el estado de ánimo de las personas varía. En este trabajo, hemos considerado únicamente la distinción entre días laborables y días de fin de semana porque las variaciones dentro de estos períodos son menores.

Se consideran días laborables a los “lunes”, “martes”, “miércoles” y “jueves”. Por lo tanto, los días de fin de semana son los “viernes”, “sábados” y “domingos”.

- **Período del año:** los períodos en los que se divide el año son los más lógicos: **las estaciones**.

En este caso, se definen las estaciones como agrupaciones de meses naturales, es decir, la “primavera” está compuesta por los meses que van de “marzo” a “mayo”; el “verano” se compone de los meses que van desde “junio” hasta “agosto”; el “otoño” se compone por el período que va desde “septiembre” hasta “noviembre”; y por último, el “invierno” va de “diciembre” a “febrero”.

Gracias a estas agrupaciones se consigue la generalización del contexto temporal.

4.3.2 Sensor de localización

En este momento, el sensor de localización consiste en presentar al usuario una serie de lugares en forma de selección para que él mismo elija dónde se encuentra.

La lista actual de ubicaciones es:

- **En el trabajo**
- **En casa**
- **En el coche/bus**
- **En un bar/discoteca**

Esta lista en sí, ya conforma una generalización del contexto de localización, por lo que no será necesario realizar ninguna otra operación de generalización en esta parte del contexto.

Es posible aumentar este sensor con opciones como “con amigos”, “con mi pareja”, “solo”, etc. ya que la compañía y las actividades que se realizan también influyen en el estado de ánimo.

En trabajos posteriores se introducirá un módulo de geolocalización (ya implementado) para una obtención de la ubicación del usuario más completa. Se hablará de este módulo en el apartado “Trabajo futuro”. En este caso, la generalización sí que será necesaria, pues gracias a la geolocalización se obtienen únicamente las coordenadas del usuario, que son únicas (porque no va a haber dos personas suficientemente juntas como para que las coordenadas sean las mismas). Nunca coincidirían dos usuarios si se tuviera en cuenta únicamente las coordenadas, por esto es tan importante la generalización.

4.3.3 Sensor de estado de ánimo

Lo ideal para la obtención del estado de ánimo es la inferencia. Hacerle preguntas al usuario (con una base psicológica) para obtener su situación anímica actual.

TENSO	PREOCUPADO	ALERTA	EXCITADO
ESTRESADO	NERVIOSO	EXALTADO	FELIZ
TRISTE	ADORMECIDO	CONTENTO	SERENO
DEPRIMIDO	FATIGADO	RELAJADO	CALMADO

Sin embargo, esto es muy costoso en tiempo, ya que no todos los usuarios están dispuestos a hacer un test psicológico para, simplemente, decir que están “contentos”. Por este motivo, hemos optado por la obtención del estado de ánimo de una forma simple y gráfica.

Se presenta un cuadro con los 16 estados de ánimo más comunes para que el usuario haga la selección del que crea más adecuado a su estado actual.

Ilustración 5 : Sensor de estados de ánimo

Los estados de ánimo se presentan al usuario en forma de cuadro. Este cuadro está distribuido siguiendo el modelo de representación de emociones de Russell, es decir, cuanto mayor es la excitación de un sentimiento, más arriba está en el cuadro.

El modelo de Russell se ha seguido para el desarrollo de la representación gráfica de los estados de ánimo de la aplicación, pero a su vez, se ha elegido una forma que fuera sencilla para el usuario y para la codificación de la adquisición del sentimiento de dicho usuario.

Por este motivo, no todos los sentimientos se ajustan exactamente a las valencias y excitaciones que describe Russell en su modelo.

4.4 Módulo de acceso a datos

Existe un módulo que gestiona el acceso a datos, esto es, un módulo independiente que se encarga de la entrada y salida de datos de la base de datos.

Este módulo se ha realizado siguiendo un patrón **DAO** (objeto de acceso a datos). Este componente sirve para suministrar una interfaz común entre el sistema y la base de datos. Gracias a esta capa, la aplicación no tiene que relacionarse directamente con la base de datos. Si en un futuro se quisiera cambiar el tipo de base de datos, la aplicación quedaría exactamente igual y lo único que habría que modificar sería el DAO.

Este módulo es independiente de la aplicación y, gracias a él, sería sencillo extender la aplicación para dar soporte a distintos tipos de bases de datos. El elemento principal del módulo es una “**factoría**”; esta factoría se encarga de gestionar y generar los objetos de acceso a datos.

La factoría se ocupa de crear las tablas de la base de datos, rellenarlas, actualizar la base de datos, eliminar las tablas cuando es necesario y dar acceso a los diferentes objetos que gestionan las entradas y salidas de las tablas.

La gestión de cada tipo de objeto se hace a través de una implementación particular de las interfaces genéricas de acceso a datos (DAOS). Por este motivo, existe una implementación para la gestión de usuarios, la gestión de contextos, la gestión de ítems, etc.

4.4.1 Gestión de usuarios

En la base de datos, la tabla de usuarios tiene los siguientes campos:

ID	NAME	PASS	ACOUSTIC	...	BLUES	...	ROCK	SOUL	...	Y90
----	------	------	----------	-----	-------	-----	------	------	-----	-----

Tabla 5 : Tabla de usuarios en base de datos

Se guarda un **id** que se crea automáticamente, el **nombre** y la **contraseña**, y las **preferencias musicales** del usuario (todos los géneros musicales predefinidos). Para indicar los gustos musicales del usuario, se guarda un 1 en la columna correspondiente. Si, por el contrario, el usuario no ha indicado que ese género musical es de su agrado, se almacena en la base de datos un 0.

El nombre de usuario y la contraseña son los proporcionados por el usuario. Estos sirven para el inicio de sesión del usuario en la aplicación. El id sirve para localizar de una manera simple al usuario en la base de datos. En nuestro caso, el id es un número y es único, es decir, no hay dos usuarios que tengan el mismo id.

La gestión de los usuarios en la base de datos se define en una clase propia, llamada “UserDAO”. En esta clase están los métodos necesarios para la **inserción**, **selección**, **eliminación**, **actualización** y para el resto de consultas necesarias para esta base de datos.

4.4.2 Gestión de ítems

Actualmente, en la aplicación se recomienda música por **artista**. El conjunto de datos que se maneja para definir un artista es:

- **Nombre** del artista.
- Dirección web con la **información** del artista.
- Dirección web con la **imagen** del artista.
- Tres direcciones web con **vídeos musicales** del artista.
- **Géneros musicales** del artista.

Para almacenar estos datos son necesarias dos tablas en la base de datos: la **tabla de artistas** y la **tabla de géneros**.

ID	NOMBRE	URL INFO	VIDEO1	VIDEO2	VIDEO3	IMAGEN
1	Lady Gaga	http://...	http://...	http://...	http://...	http://...

Tabla 6 : Tabla de artistas en base de datos

En la tabla de artistas, se guardan el id, creado automáticamente, el nombre del artista, una dirección web con información, una dirección web con una imagen y tres direcciones más con vídeos musicales del artista. Los datos utilizados en este trabajo provienen de un estudio relacionado con emociones en recomendación [4].⁴

Esta tabla se complementa con la tabla de géneros, en la cual se almacenan los géneros del artista con unos pesos determinados.

NAME	ACOUSTIC	...	ELECTRONIC	...	POP	...	Y80	Y90
Lady Gaga			0.35		0.5			0.15

Tabla 7 : Tabla de géneros en base de datos

La gestión de los artistas en la base de datos se define en la clase "ItemDAO". En esta clase se definen los métodos necesarios para la **inserción**, **selección**, **actualización** y **eliminación** de los artistas y las canciones en la base de datos, además de otros métodos útiles para la gestión de estos datos.

⁴ EC-Web, <http://ir.ii.uam.es/emotions> - 5/VII/2014

4.4.3 Gestión de contextos

En la base de datos se guardan los contextos normalizados de los usuarios. Un contexto tiene los siguientes atributos en la base de datos:

- **Id**
- **Estado de ánimo**
- **Ubicación**
- **Año**
- **Período del día**
- **Período de la semana**
- **Período del año**

ID	MOOD	PLACE	YEAR	PDAY	PWEEK	PYEAR
1	Happy	Bar	2014	Morning	Weekend	Spring

Tabla 8 : Tabla de contextos en base de datos

El contexto se guarda con todos sus atributos en la base de datos; algunos obtenidos gráficamente por la selección del usuario y otros mediante la marca de tiempo.

La gestión de los contextos normalizados en la base de datos se define en la clase “ContextDAO”. En esta clase se han implementado los métodos necesarios para la **inserción, selección, actualización y eliminación** de los contextos en la base de datos, además de otros métodos útiles para gestionar los contextos.

4.4.4 Gestión de interacciones

Las interacciones en nuestra base de datos están formadas por **ratings** y **cambios de estado de ánimo**.

La clase que gestiona las interacciones se llama “InteractionDAO”. En esta clase se encuentran los métodos necesarios para la **inserción, actualización, selección y eliminación** en base de datos de ratings y cambios de estado de ánimo. Además, se incluyen otros métodos útiles para la gestión de las interacciones.

Existen también métodos ya implementados para la gestión de “tags” (anotaciones del usuario). Estos métodos serán utilizados para una mejora de la aplicación. Esta mejora se explicará en el apartado “Trabajo futuro”.

Ratings

Llamamos “**rating**” a la puntuación que le da un usuario a un artista. Es importante el almacenamiento de las puntuaciones porque gracias a ellas se puede medir de una forma numérica la exactitud del sistema de recomendación. Si el usuario califica uno de los artistas recomendados con una puntuación alta, significa que ese artista le ha gustado y que, por tanto, la recomendación ha sido satisfactoria.

Los ratings se componen de 5 atributos:

ID	ID_USER	ID_ARTIST	ID_CONTEXT	RATING
1	3	4	1	0
2	2	4	2	5

Tabla 9 : Tabla de ratings en base de datos

Se guardan los ids del **usuario** que puntúa y **artista** al que se le concede la puntuación; además, se almacena el id del **contexto** en el que se ha puntuado al artista y, obviamente, la **puntuación** que se le ha dado. El id del rating se genera automáticamente.

Las puntuaciones disponibles van de 0 a 5 estrellas.

Cambios de estados de ánimo

Una de las mediciones de la aplicación es la monitorización de los cambios de estado de ánimo que se producen a través de la música en un cierto contexto. Por esto, la tabla de cambios de estado de ánimo es muy importante en la aplicación.

ID	ID_USER	ID_ARTIST	ID_CONTEXT	FINAL MOOD
1	2	5	1	RELAJADO
2	3	4	2	ADORMECIDO

Tabla 10 : Tabla de cambios de estado de ánimo en la base de datos

Gracias a esta tabla, se obtienen todos los componentes necesarios para la monitorización del cambio de estados de ánimo. El **usuario**, el **artista**, el **contexto** (donde se encuentra también el estado de ánimo inicial) y el **estado de ánimo final**, que se obtiene al escuchar una de las canciones del artista.

4.5 Módulo de recomendaciones

Para obtener las recomendaciones musicales, se procesan los datos de los usuarios y de los artistas y se combinan para obtener una lista ordenada de artistas.

La puntuación de la recomendación de los artistas se obtiene mediante un **post-filtrado contextual**, es decir, en un primer momento se obtiene la puntuación del artista teniendo en cuenta únicamente la similitud con el usuario (los géneros musicales que tienen en común). Una vez obtenida la puntuación basada en contenido, se calcula la similitud del artista con el contexto del usuario. Cuando se tienen las dos puntuaciones, se combinan de la siguiente manera:

$$R(u, a, u_c) = \lambda \cdot CB(u, a) + (1 - \lambda) \cdot CTX(a, u_c)$$

donde:

- $R(u, a, u_c)$: puntuación de un artista para un usuario en un contexto concreto.
- λ : lambda
- u : usuario
- a : artista
- u_c : contexto del usuario
- $CB(u, a)$: puntuación de la recomendación basada en contenido
- $CTX(a, u_c)$: puntuación de la recomendación basada en contexto

La fórmula de la recomendación se compone de dos partes: **recomendación basada en contenido** (CB, content based) y **recomendación basada en contexto** (CTX).

La unión de ambas partes es el parámetro λ (lambda), que pondera la ecuación dando más peso a la recomendación basada en contenido o a la recomendación basada en contexto.

La llamada para obtener las recomendaciones es la siguiente:

```
//Creacion del objeto recomendador
Recommender rec= new Recommender(this, lambda_rec);

//Obtencion de los artistas a recomendar
List<MusicArtist> list_artists = rec.recommend(context, user);
```

Código 1 : Llamada al objeto recomendador

Se crea el objeto **recomendador** y se le pasa el contexto Android de la aplicación (necesario para el acceso a la base de datos) y el parámetro “**lambda**” que ponderará las puntuaciones de los artistas.

A continuación se hace la llamada al método “recomendar” para obtener una lista ordenada de artistas, que se corresponderán con las recomendaciones. Este método necesita el contexto normalizado de la transacción (formado por los datos obtenidos de los sensores) y el usuario al que se le va a recomendar.

Se explicarán las dos partes de la recomendación en los siguientes apartados.

4.5.1 Módulo de filtrado basado en contenido

La recomendación personalizada es la basada en contenido, ya que en esta parte de la recomendación se comparan los géneros preferidos del usuario con los géneros que toca un artista.

$$CB(u, a) = \cos(u, a) = \frac{u \cdot a}{\|u\| \cdot \|a\|} = \frac{\sum_k u_k \cdot a_k}{\sqrt{\sum_k u_k^2} \cdot \sqrt{\sum_k a_k^2}}$$

donde:

- **CB(u,a)**: puntuación de la recomendación basada en contenido
- **cos(u,a)** : similitud cosenoidal entre el usuario **u** y el artista **a**
- **u**: usuario
- **a**: artista
- **u_k**: género musical del usuario
- **a_k**: género musical del artista

En esta recomendación, se considera tanto al usuario como al artista como un **vector**. Estos vectores están formados por géneros musicales.

	Pop	Rock	Clásica	Punk
Usuario	0	1	1	0
Artista	0	0.7	0	0.3

Tabla 11 : Ejemplo de vectorización de usuarios y artistas

En este ejemplo, el vector usuario sería $\vec{u} = (0, 1, 1, 0)$. El vector artista, por el contrario, sería $\vec{a} = (0, 0.7, 0, 0.3)$.

La **similitud cosenoidal** evalúa el valor del coseno del ángulo comprendido entre los dos vectores. La función trigonométrica produce un valor de 1 (el máximo) si el ángulo que existe entre los dos vectores es cero, es decir, si los dos vectores apuntan en el mismo sentido de una dirección.

Esta métrica devuelve **valores entre 1 y -1** en un intervalo cerrado. Por este motivo, esta similitud es muy utilizada en minería de datos.

En este trabajo se ha utilizado esta similitud porque es muy fácil transformar los datos que utiliza la aplicación en vectores. En el caso de los usuarios, son vectores que se componen de ceros y unos, indicando que géneros son interesantes para el usuario. En el caso de los artistas, son vectores que se componen de valores reales positivos que indican el grado de similitud del artista con el género, es decir, si ese artista toca ese género.

La implementación del algoritmo se consigue haciendo la similitud basada en coseno entre los vectores de usuarios y artistas. Esto es, multiplicar los vectores (los géneros del usuario por los del artista) y dividir entre la multiplicación de sus módulos.

```
/**
 * Recomendacion basada en contenido
 * @param user usuario
 * @return HashMap<MusicArtist, Double> mapa que contiene todos los artistas
 * con sus puntuaciones
 * Se cruzan los valores de los generos del usuario con los del artista.
 *  $CB(u,a)=(u*a)/(|u||a|)$ 
 */
public HashMap<MusicArtist, Double> recommendCB(UserProfile user){

    double moduser = 0;
    double modartist = 0;
    double total = 0;

    //obtencion de todos los artistas de la base de datos
    ArrayList<MusicArtist> artists = itemDAO.getAllArtist();

    //mapa donde se almacenaran los artistas con sus puntuaciones
    HashMap<MusicArtist, Double> map = new HashMap<MusicArtist, Double>();

    for(MusicArtist artist : artists){

        //El usuario tiene todos los generos indicando si le gustan (1) o si
        //no (0)
        //El artista solo tiene los generos que toca
        for(String genre : artist.getGenres().keySet()){
            //obtencion del total y del modulo del artista
            total += artist.getGenres().get(genre)*user.getGenres().get(genre);
            modartist += Math.pow(artist.getGenres().get(genre), 2);
        }

        for(String genre : user.getGenres().keySet()){
            //obtencion del modulo del usuario
            moduser += Math.pow(user.getGenres().get(genre), 2);
        }

        //si el total no es cero se introduce el artista y la puntuacion en el
        //mapa
        if(total!=0)
            map.put(artist,total/(Math.sqrt(moduser)*Math.sqrt(modartist)));

        total = 0;
        modartist = 0;
        moduser = 0;
    }

    //normalizacion de las puntuaciones de los artistas
    if(!map.isEmpty())
        return normalize(map);
    return map;
}
```

Código 2 : Implementación del algoritmo de recomendación basado en contenido

Se multiplican todos los géneros definidos en el artista por los del usuario. Si no son comunes, el usuario pondrá ese valor a cero. Y a la vez, se calcula el sumatorio del módulo del artista.

A continuación, se calcula el sumatorio del módulo del usuario y se añade la puntuación total a un HashMap, que se devolverá normalizado⁵.

4.5.2 Módulo de filtrado contextual

El módulo de recomendaciones contextualizadas se encarga de obtener la lista ordenada de ítems a recomendar al usuario en base al contexto de la transacción.

$$CTX(a, u_c) = \sum_c s(c, u_c) * s(a, c)$$

donde:

- **CTX(a, u_c)**: puntuación de la recomendación basada en contexto
- **s(c, u_c)**: similitud del contexto **c** con el contexto del usuario **u_c**
- **s(a, c)**: similitud del artista **a** con el contexto **c**
- **a**: artista
- **u_c**: contexto del usuario
- **c**: contexto

La puntuación que obtiene un artista en un contexto (puntuación basada en contexto) se obtiene gracias al sumatorio de dos partes de una ecuación: la **similitud del todos los contextos** con el contexto actual, el contexto del usuario, y la **similitud del artista con todos los contextos**.

Similitud de contextos:

$$s(c, u_c) = sim(c, u_c)$$

Con esta parte de la ecuación se calcula la similitud del contexto a comparar con el contexto del usuario. En nuestro caso, se compara el contexto actual con todos los contextos almacenados en la base de datos.

Puntuación de un artista en un contexto:

$$s(a, c) = \frac{1}{D} \sum_u r(u, a, c)$$

donde:

- **s(a, c)**: similitud del artista **a** con el contexto **c**
- **a**: artista
- **c**: contexto
- **u**: usuario
- **r(u, a, c)**: puntuación que el artista **u** asignó al usuario **a** en el contexto **c**
- **D** = $\sum_a \sum_u r(u, a, c)$; **D** es la puntuación total de los ítems (artistas) que hay en un contexto.

⁵ La normalización de las puntuaciones se explicará en el apartado 4.5.3

En este caso, se ha optado por una estrategia de puntuación muy sencilla. Se halla la **puntuación promedio del artista**. Para poder obtener este promedio, es necesario calcular la puntuación del artista en todos los contextos.

Para poder interiorizar el mecanismo de recomendación, es imprescindible comprender como se realiza la **similitud de contextos**.

Para la recomendación basada en contexto es necesario comparar contextos. Se debe obtener la similitud de unos contextos con otros y también la similitud de los contextos con los ítems. Estos valores se utilizarán para la recomendación de artistas.

En nuestro caso, la generalización de contextos se realiza antes de su inserción en la base de datos, para que, posteriormente, la comparación de contextos sea más sencilla.

Como los campos en base de datos de los contextos permiten únicamente un número finito de posibilidades (16 estados de ánimo, 4 lugares...), la similitud se realiza comparando los campos.

Por ejemplo, si tuviéramos que obtener la similitud de estos dos contextos:

ID	MOOD	PLACE	YEAR	PDAY	PWEEK	PYEAR
1	Alerta	Coche	2014	Morning	Weekend	Spring
2	Sereno	Trabajo	2014	Morning	Weekday	Spring

Tabla 12 : Ejemplo de contextos en base de datos

La similitud que se obtendría tendría un valor de 3, ya que los únicos campos que tienen en común los dos contextos son el año, el período del día y el período del año. La mayor similitud es 6 (ya que este es el número de campos que tiene un contexto sin contar el id, que se genera automáticamente).

La base de datos de contextos se ha creado de tal forma que cada contexto en la base de datos es único, es decir, no hay dos contextos insertados en la base de datos cuya similitud sea 6. Antes de la inserción en base de datos de un contexto nuevo, se compara con los insertados previamente. Si coinciden todos los campos, es decir, si la similitud es máxima (6 en nuestro caso), el contexto no se inserta en la base de datos y se obtiene el id del contexto igual ya existente en la base de datos. Si, por el contrario, no existe ningún contexto ya insertado cuya similitud con el actual es máxima, se inserta el nuevo contexto.

La similitud de contextos se constituye como una parte imprescindible de este algoritmo de recomendación.

```

/**
 * Recomendacion basada en contexto
 * @param context contexto del usuario
 * @return HashMap<MusicArtist, Double> mapa que contiene todos los artistas
 con sus puntuaciones
 * CTX(a,c)= Sumatorio [p(c|u)*p(a|c)]
 */
public HashMap<MusicArtist, Double> recommendCTX(NormalizedContext context){

    //obtencion de todos los contextos y los artistas de la base de datos
    ArrayList<NormalizedContext> cntxs = contextDAO.getAllContext();
    ArrayList<MusicArtist> artists = itemDAO.getAllArtist();

    //mapa donde se guardaran los artistas con sus puntuaciones
    HashMap<MusicArtist, Double> map = new HashMap<MusicArtist,Double>();

    if(cntxs == null)
        return null;

    for(MusicArtist artist : artists){

        //Para cada contexto se multiplica la similitud con el contexto actual
        y se halla la similitud del item con el contexto.
        for(NormalizedContext c : cntxs){
            //si la similitud es cero no se introduce el artista en el mapa
            if((context.getSimilarity(c) == 0) || (simItemContext(c,artist)==0)){
                continue;
            }
            map.put(artist, (context.getSimilarity(c)*simItemContext(c,artist)));
        }
    }

    //normalizacion de las puntuaciones del mapa
    if(!map.isEmpty())
        return normalize(map);
    return map;
}

```

Código 3 : Implementación del algoritmo de recomendación basado en contexto

La función de recomendación basada en contexto recibe como parámetro un contexto normalizado. Para cada artista presente en la base de datos, se halla su puntuación obteniendo la similitud con todos los contextos (similitud de ítem con contexto, explicada a continuación) y la similitud de todos los contextos con el contexto del usuario actual.

Una vez obtenidos estos valores, se multiplican los pares de datos y se introduce el valor resultante en un HashMap, junto al artista, que posteriormente se normalizará.


```

/**
 * similitud de un item con un contexto
 * @param context contexto del usuario
 * @param artist artista
 * @return puntuacion de la similitud
 */
public Double simItemContext(NormalizedContext context, MusicArtist artist){

    double total_item = 0;
    double total = 0;

    //HashMap<Long, Long[]>
    //HashMap<ID_RATING, [ID_USER, ID_ARTIST, ID_CONTEXT, RATING]>
    HashMap<Long, Long[]> ratings =
        interactionDAO.getAllRatingsinContext(context);

    if(ratings.isEmpty())
        return total;

    //Se obtiene la puntuacion total de ese item en el contexto
    //(total_item) y la puntuacion de todos los items puntuados en ese
    //contexto (total)
    for(Long id : ratings.keySet()){
        if(ratings.get(id)[1] == artist.getID()){
            total_item += ratings.get(id)[3];
            total += ratings.get(id)[3];
        }
    }

    //Se devuelve la division del total del item en el contexto por la
    //puntuacion de todos los items en ese contexto
    return total_item/total;
}

```

Código 4 : Similitud de un ítem con un contexto

Para obtener la similitud de un artista con un contexto se obtiene la puntuación promedio del artista, es decir, la división del total del artista en el contexto y la puntuación total de todos los ratings del contexto.

La implementación del algoritmo se consigue en un único bucle, donde se calculan ambas medidas (el total del artista y el total del contexto). Como se puede observar en la imagen, se devuelve la división de los dos valores.

4.5.3 Normalización de puntuaciones

Para poder unir las dos partes de la recomendación (recomendación basada en contexto y recomendación basada en contenido), es importante que ambas puntuaciones estén en el **mismo rango**, es decir, que estén en escalas comparables. En este trabajo, esto se hace mediante un **método de normalización** de puntuaciones.

$$pn = \frac{x - \min}{\max - \min}$$

donde:

- **pn**: puntuación normalizada
- **x**: puntuación del elemento sin normalizar
- **min**: mínimo de todo el conjunto
- **max**: máximo de todo el conjunto

```
/**
 * normalizacion del mapa de puntuaciones de artistas
 * @param map mapa de artistas y puntuaciones
 * @return HashMap<MusicArtist, Double> puntuacion normalizada del mapa
 * Normalizacion x = (x - min) / (max - min)
 */
public HashMap<MusicArtist, Double> normalize(
    HashMap<MusicArtist, Double> map) {

    HashMap<MusicArtist, Double> normalize = new
        HashMap<MusicArtist, Double>();

    double min = (Collections.min(map.values()));
    double max = Collections.max(map.values());

    //Si el maximo es igual al minimo, todos los
    //artistas obtienen una puntuacion de 1.0
    if(min == max){
        for(MusicArtist a : map.keySet()){
            normalize.put(a, 1.0);
        }
        return normalize;
    }

    //Formula de normalizacion
    for(MusicArtist a : map.keySet()){
        normalize.put(a, (map.get(a) - min)/(max-min));
    }

    return normalize;
}
```

Código 5 : Normalización de puntuaciones

En el caso de la normalización, se comprueba si el HashMap a normalizar está vacío, si no lo está, se procede a aplicar la fórmula a todos los elementos del conjunto. En primer lugar se obtiene el máximo y el mínimo y a continuación, se normalizan todas las entradas del grupo.

Aplicación móvil desarrollada

En esta sección se describe la aplicación móvil desarrollada. Se explican las tecnologías empleadas en la misma y se detallan las pantallas principales de la interfaz gráfica. Además, se explica la evaluación realizada a la aplicación.

5.1 Tecnologías empleadas

Este proyecto está compuesto por **módulos** que combinan distintas tecnologías y lenguajes de programación. La mayor parte de la aplicación está implementada en el lenguaje **Java**. Existe un módulo, la interfaz de usuario, que está implementado utilizando las librerías de **Android 4.4.2**.

La base de datos utilizada es **SQLite**, la base de datos que se utiliza por defecto en Android. Se consideró usar un sistema de ficheros y también otros tipos de bases de datos, pero estas ideas se descartaron porque la implementación del almacenamiento de datos en bases de datos SQLite es mucho más asequible en combinación con Android.

Todo el proyecto está desarrollado en **Eclipse SDK**, ya que en esta plataforma, la integración de los componentes y librerías de Android es sencilla y eficaz. La aplicación se ha ido desarrollando usando un emulador, **dispositivo virtual** (instalado en Eclipse) y varios **dispositivos físicos** (Nexus 4, Moto G, Cubot...).

Se han hecho pruebas de visualización tanto en dispositivos físicos (teléfono, tablet...) como en el emulador de Eclipse. Para el desarrollo de la aplicación se ha usado más el dispositivo físico que el emulador, porque el emulador es más lento y, al final, la aplicación será usada en un dispositivo físico. Por estos motivos, se han empleado más los dispositivos físicos en las pruebas y en la evaluación de la aplicación.

Es importante destacar los tipos de pantalla empleados para implementar los componentes de la aplicación. En Android, los tipos de pantalla se definen en los “layouts”. El tipo principal de pantalla que se ha utilizado es el “**LinearLayout**”. Este layout combinado con otros elementos conforma la totalidad de las pantallas de la aplicación.

Gracias al empleo de estas tecnologías, la aplicación resultante es tanto fácil de usar como de comprender a nivel programador. La sencillez de las clases y la comodidad de los elementos específicos de Android hacen que sea muy simple interpretar la arquitectura y el flujo de la aplicación.

5.2 Componentes principales de la interfaz gráfica

La presentación de la aplicación al usuario se realiza por medio de pantallas. Estos componentes gráficos son muy accesibles al usuario y muy fáciles de utilizar.

Los componentes principales de la interfaz gráfica de la aplicación se explicarán en los siguientes apartados.

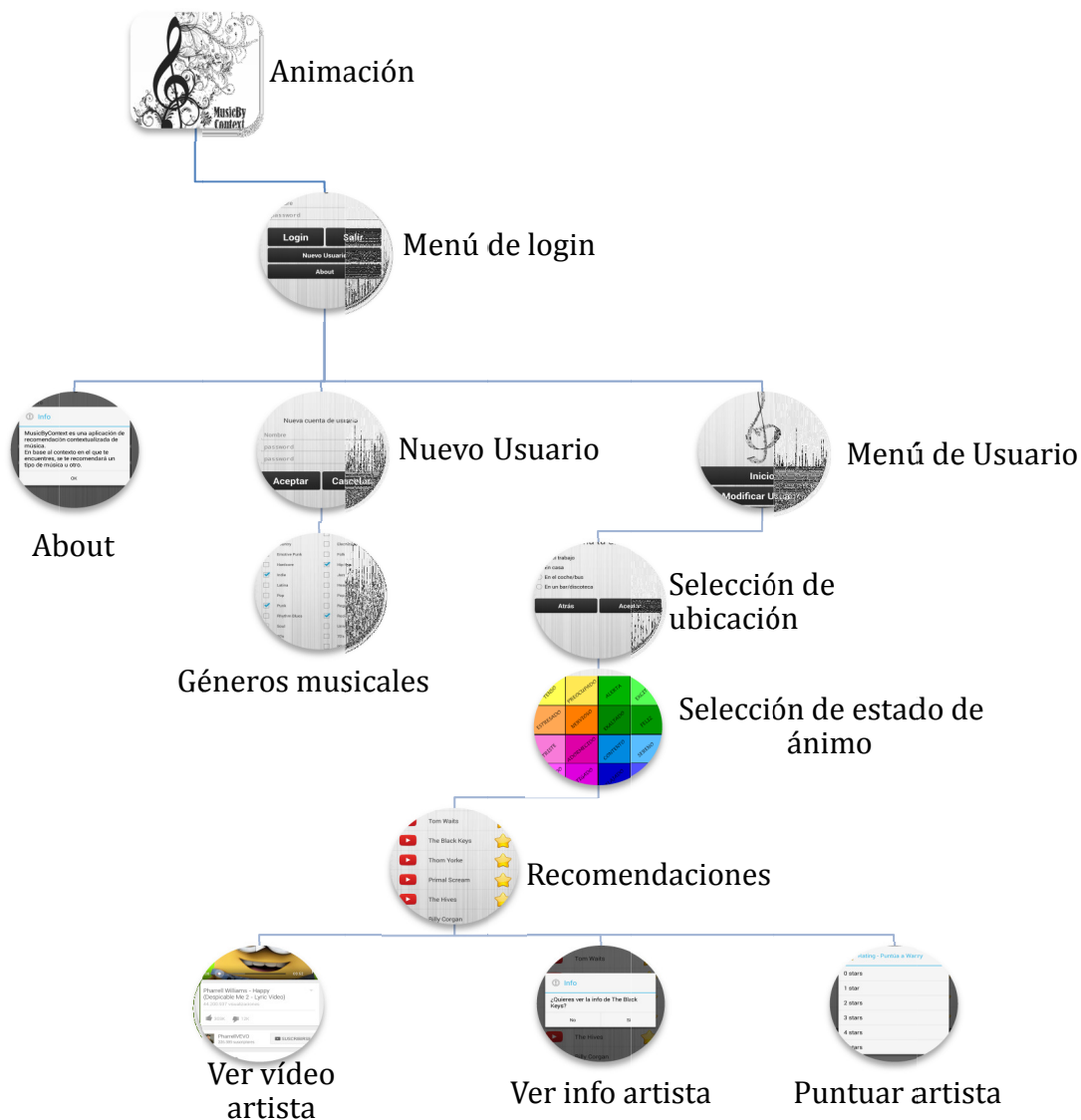


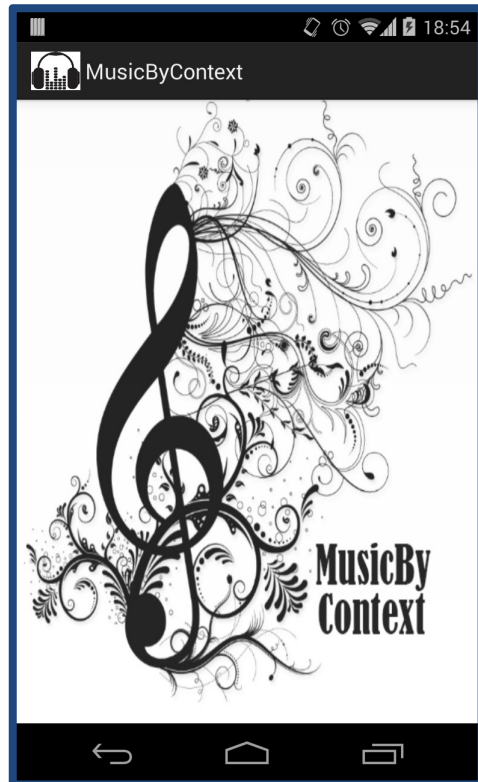
Ilustración 6 : Diagrama de transiciones de pantallas de la aplicación

5.2.1 Animación de presentación

Esta pantalla es la **introducción** a la aplicación. Se muestra al abrir la aplicación y es un nexo entre el usuario y el menú principal.

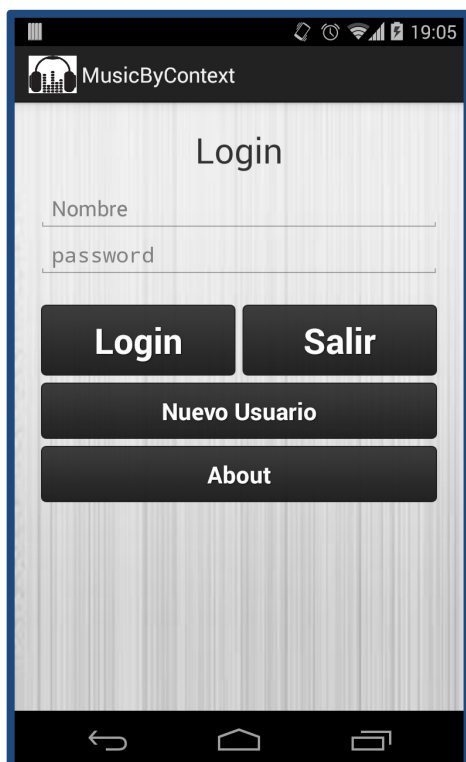
Se muestra una imagen y se crea el efecto de “aparición”. La pantalla comienza en blanco y la imagen se va haciendo visible hasta llegar a ser nítida. A continuación, se pasa al menú de login.

Ilustración 7 : Pantalla de animación



5.2.2 Menú de login

Esta pantalla se encarga de manejar el **acceso del usuario a la aplicación**.



El acceso se gestiona mediante la inserción de un nombre de usuario y contraseña correctos y la pulsación del botón de **Login**. Este botón conducirá al usuario al menú de usuario.

El botón **Salir** permite al usuario salir de la aplicación.

El botón **Nuevo Usuario** permite crear un usuario nuevo en la base de datos.

El botón **About** muestra al usuario la información de la aplicación en un diálogo.

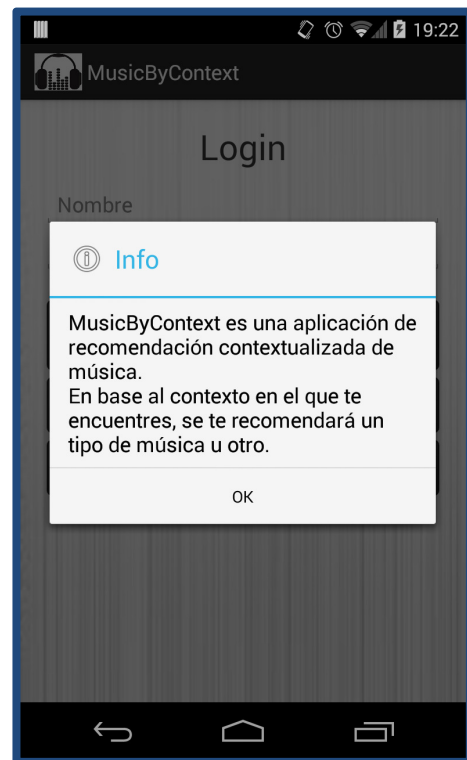
Ilustración 8 : Pantalla del menú de login de la aplicación

5.2.3 About

El botón **About** de la pantalla de **Login** infla un cuadro de diálogo que muestra la información de la aplicación.

Esta información es muy sencilla. Únicamente se muestran un par de párrafos con una breve descripción de la aplicación.

Ilustración 9 : Diálogo de información de la aplicación



5.2.4 Nuevo Usuario

Esta pantalla permite crear un usuario nuevo e insertarlo en la base de datos. Se llega a esta pantalla mediante el botón **Nuevo Usuario** que se encuentra en la pantalla de **Login**.

Los datos del usuario que se piden son el nombre y la contraseña. La contraseña se repite dos veces para que el usuario se asegure de que la ha introducido correctamente.

Al pulsar el botón **Aceptar** se le permite al nuevo usuario seleccionar los géneros musicales que le gustan.

Ilustración 10 : Pantalla de creación de un nuevo usuario

5.2.5 Selección de géneros musicales

En esta pantalla se muestran los **géneros musicales** disponibles para que el usuario seleccione sus preferencias.

Se muestran los géneros musicales con un “CheckBox” a su derecha, el cual simboliza la selección del género musical.

Una vez seleccionados, se insertará el nuevo usuario en la base de datos.

Se pueden seleccionar tantos géneros como desee el usuario.

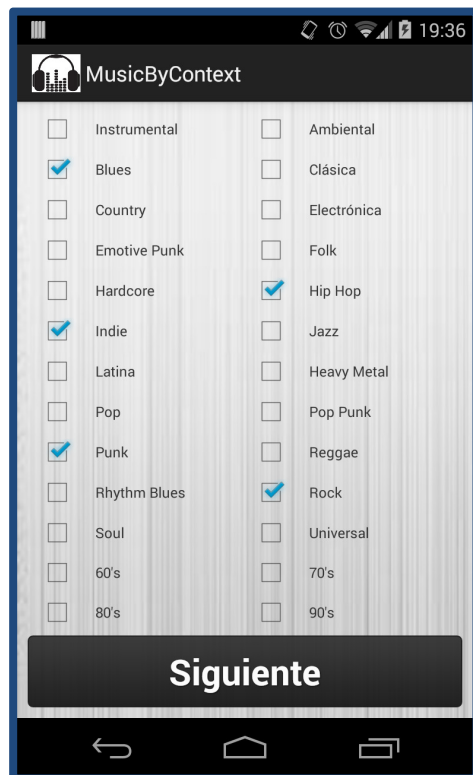


Ilustración 11 : Pantalla de selección de géneros musicales

5.2.6 Menú de usuario



Este es el **menú principal del usuario**. Una vez logueado, el usuario viene aquí.

El botón **Inicio** sirve para iniciar la recomendación de música. Después se le pedirá al usuario que introduzca su estado de ánimo y su ubicación.

El botón **Modificar Usuario** permite al usuario cambiar de nombre, de contraseña y cambiar también sus preferencias musicales.

El botón **Salir** lleva al usuario al menú principal de la aplicación, la pantalla de **Login**.

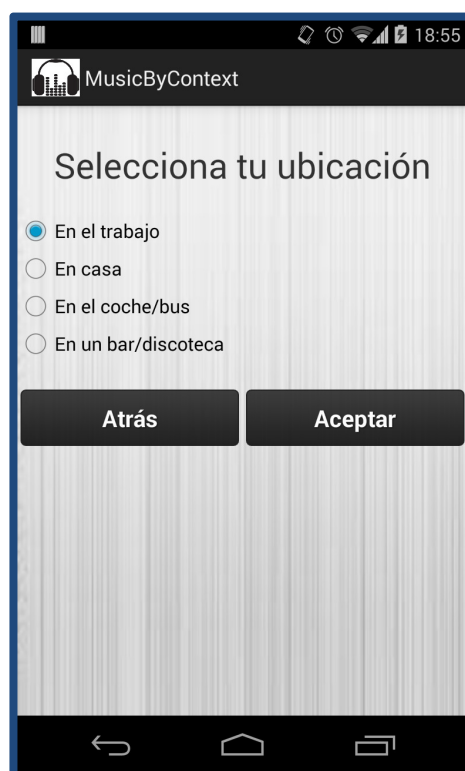
Ilustración 12 : Pantalla del menú de usuario de la aplicación

5.2.7 Selección de ubicación

En esta pantalla el usuario deberá seleccionar su **ubicación**. Se muestran las cuatro opciones ya comentadas en este documento en forma de “RadioGroup”.

El “RadioGroup” limita la selección del usuario a un único botón, al contrario que los “CheckBox” utilizados en la selección de géneros musicales.

Ilustración 13 : Pantalla de selección de localización



5.2.8 Selección de estado de ánimo



En esta pantalla se ofrece al usuario una selección de **16 estados de ánimo** que se corresponden con los representados en el modelo de Russell. Son los estados de ánimo más comunes que cualquier ser humano ha sentido en algún momento.

Al seleccionar el estado de ánimo, aparece un mensaje en la pantalla diciendo cuál se ha seleccionado.

Al pulsar el botón **Recomendar** se calculan las puntuaciones de los artistas de la base de datos y se muestran en la siguiente pantalla.

Ilustración 14 : Pantalla de selección de estado de ánimo

5.2.9 Lista de recomendaciones

Se ofrece al usuario una **lista de recomendaciones con 15 artistas**. Estos artistas son los que, al aplicar el algoritmo de recomendación, han obtenido una puntuación mayor.

La lista de artistas aparece ordenada en la pantalla: el artista con mayor puntuación aparece el primero de la lista, el que ha obtenido la segunda puntuación mayor se muestra el segundo y así sucesivamente.

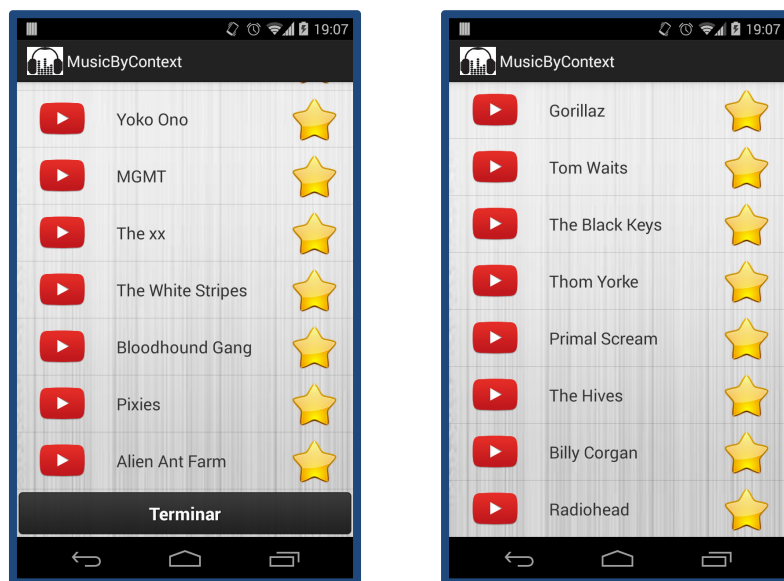
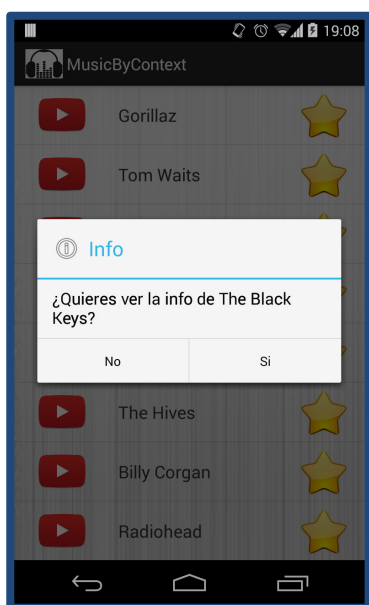


Ilustración 15 : Pantallas de recomendaciones

Al pulsar el botón **Terminar** se da por concluida la recomendación y se vuelve al menú principal de usuario.



En esta pantalla existen otros tres botones “camuflados”. El icono de **Play**, el **nombre del artista**, y la **estrella**.

El icono de **Play** lleva al usuario a un **vídeo musical del artista** en Internet.

El **nombre del artista** despliega un “AlertDialog” donde se pregunta al usuario si desea ver la información del artista seleccionado. Si el usuario pulsa “Sí” se le lleva a la web del artista en “lastfm”. Si, por el contrario, el usuario pulsa “No” desaparece el “AlertDialog” y se vuelven a mostrar las recomendaciones del usuario.

Ilustración 16 : Diálogo de información del artista

El único botón que queda es el más importante de la aplicación: la **estrella**.

El botón de la **estrella** permite al usuario **puntuar al artista seleccionado** (añadir un rating en la base de datos). Las puntuaciones disponibles están entre 0 y 5.

Una vez el usuario ha seleccionado la puntuación que desea darle al artista, se inserta el rating en la base de datos.

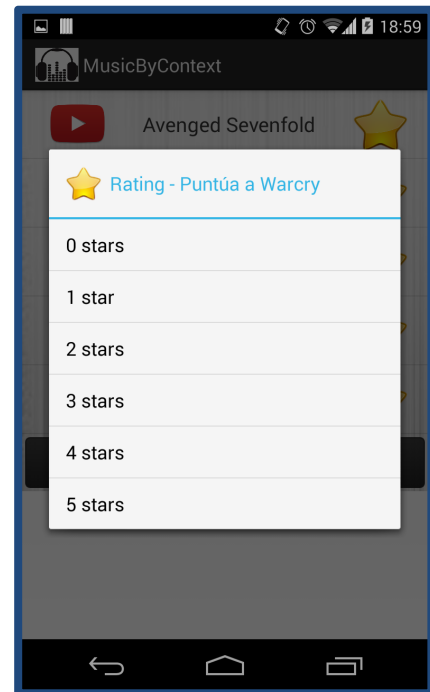
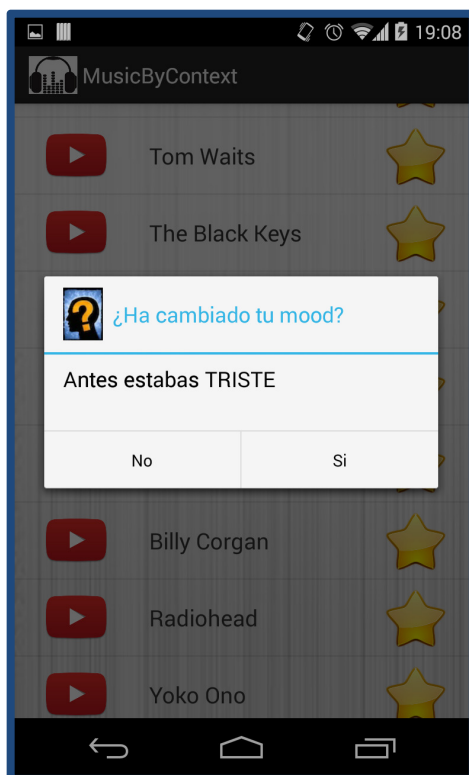


Ilustración 17 : Pantalla de puntuación a un artista



Una vez insertado el rating en la base de datos, se le pregunta al usuario si su estado de ánimo ha cambiado mediante un “AlertDialog”.

Si el usuario indica que su **estado de ánimo ha cambiado**, se le muestra otra vez la imagen con los estados de ánimo y se almacena en la tabla “ChangeMoods” de la base de datos.

Si, por el contrario, el usuario pulsa “No”, indicando que no se ha modificado su estado de ánimo, se inserta en la tabla de “ChangeMoods” con el mismo estado de ánimo que tenía anteriormente.

Ilustración 18 : Diálogo de cambio de estado de ánimo

5.3 Flujo de la aplicación

En este apartado se muestra el diagrama de flujo de la aplicación. El flujo comienza en “Login”, que se corresponde con el menú principal de la aplicación (la pantalla de “Login”) que viene justo después de que finalice la animación.

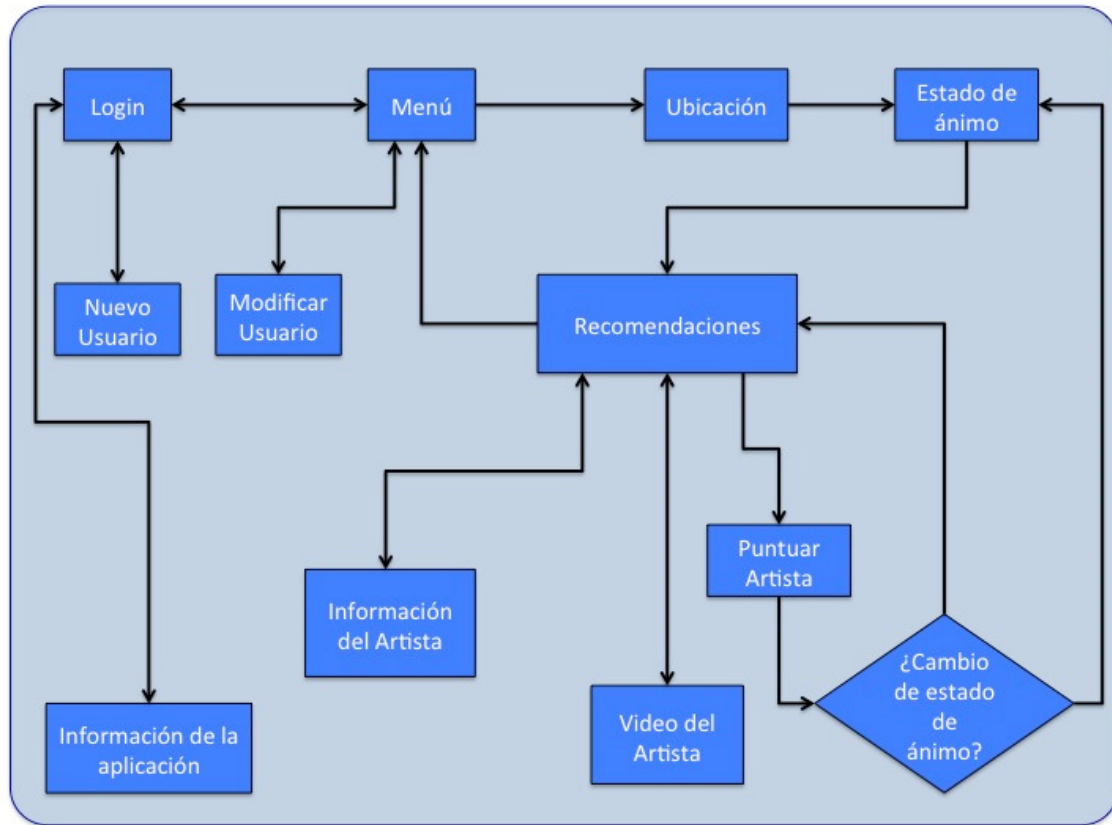


Ilustración 19 : Diagrama de flujo de la aplicación

Las flechas muestran el sentido del flujo. Las cajas se corresponden con las pantallas principales de la aplicación y el rombo representa la posible decisión del usuario cuando se le pregunta si su estado de ánimo ha cambiado. Si efectivamente ha cambiado, se le pregunta de nuevo su estado de ánimo actual y se almacena el cambio en base de datos. Si, por otro lado, su estado de ánimo no ha cambiado, se almacena este dato en la base de datos.

Gracias a esta imagen se puede comprender mejor el orden de las pantallas de la aplicación y, por tanto, se pueden interpretar más fácilmente los pasos que debe dar un usuario para poder obtener una lista de recomendaciones.

5.4 Evaluación preliminar

Para probar el correcto funcionamiento de la aplicación se ha llevado a cabo un experimento con usuarios reales. El objetivo de esta evaluación es **concluir si los usuarios encuentran útiles las recomendaciones y si aquellas basadas en contexto son mejores que las que solo tienen en cuenta las preferencias del usuario**.

En esta sección se describe la evaluación realizada sobre la aplicación desarrollada. A continuación se detalla el experimento realizado, posteriormente se explica la metodología empleada y el conjunto de datos sobre los que se ha aplicado y después se exponen los resultados de la misma.

5.4.1 Diseño del experimento

En esta evaluación, se han obtenido diversos datos de artistas y se han añadido en la base de datos de la aplicación. Concretamente, se ha incluido el nombre de los artistas, tres enlaces a vídeos musicales, una dirección para ampliar información y una lista de géneros musicales relacionados.

Para el procedimiento de evaluación de la aplicación, se ha conseguido la participación de **15 personas**. Tras registrarse en la aplicación, seleccionaban sus géneros musicales preferidos y, a continuación, introducían su ubicación actual y su estado de ánimo, para determinar así el contexto del usuario.

Una vez establecido el contexto, el sistema calcula una **lista de 15 artistas** sugeridos que se presentan al usuario. En esta lista, se permite ver uno o más vídeos musicales e información adicional de cada artista recomendado, y, opcionalmente, puntuar a dicho artista.

En un caso típico, el usuario veía y escuchaba un vídeo musical de un artista de la lista, y a continuación lo puntuaba. Cada usuario podía puntuar a tantos artistas como quisiera y las veces que deseara. Es decir, en ningún momento se ha instado a los usuarios a puntuar a un determinado número de artistas.

La puntuación que un usuario le puede dar a un artista es un número entero entre 0 y 5. En todo momento se le ha dejado libertad para poner la puntuación que deseara, es decir, no se ha animado a ningún usuario a poner una puntuación u otra. Una vez que los usuarios terminaban de puntuar una lista, tenían la opción de dejar de usar la aplicación, o volver a comenzar para obtener otra lista de recomendaciones.

5.4.2 Metodología de la evaluación

Una vez obtenidas las puntuaciones de los usuarios a los artistas, se tienen que evaluar. Para medir la calidad de las listas de recomendaciones utilizamos varias métricas basadas en ranking [9].

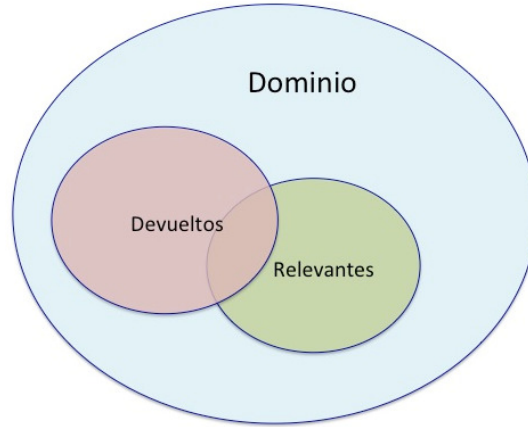


Ilustración 20 : Dominio de datos en un sistema de recomendación

En esta imagen se muestra el dominio de ítems de un sistema de recomendación.

Al ejecutar un algoritmo de recomendación, se devuelve una lista ordenada de ítems por relevancia para el usuario. Los ítems relevantes son aquellos que el usuario encuentra útiles en el contexto en el que se encuentra.

En este trabajo se han utilizado las siguientes métricas para medir la calidad de las recomendaciones:

- **Precisión.** Es la proporción de ítems relevantes entre los ítems devueltos.

$$P = \frac{| \text{Relevantes} \cap \text{Devueltos} |}{| \text{Devueltos} |}$$

De esta forma, cuanto más se acerque a 0 el valor de la precisión, mayor será el número de ítems devueltos que no se consideren relevantes.

- **Exhaustividad** o *recall*. Proporción de ítems relevantes recuperados, comparado con el total de los ítems relevantes para el usuario.

$$R = \frac{| \text{Relevantes} \cap \text{Devueltos} |}{| \text{Relevantes} |}$$

Un valor de recall igual a 1 significa que se han devuelto todos los ítems relevantes existentes para el usuario. Esta medida es apropiada para tareas en las que el usuario quiere encontrar todos los ítems relevantes y no repara en el esfuerzo de examinarlos todos.

- **Media armónica.** Es la combinación de precisión y recall. Esta métrica penaliza valores muy bajos en una u otra métrica.

$$H = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P + R}$$

donde:

- **P** : valor de precisión
- **R** : valor de recall

Es común tomar las métricas a “**k**”. Esto significa hacer un corte en los documentos devueltos a distancia “**k**” del principio de la lista. Gracias a esto se puede valorar el top de los ítems devueltos.

$$m@k = \frac{| \text{Relevantes en top } k |}{k}$$

donde **m@k** simboliza una métrica a **k**.

En nuestro caso, utilizaremos métricas **@5**, **@10** y **@15**, ya que las listas de artistas devueltas constaban de 15 artistas.

Es imprescindible definir qué es un ítem relevante. En nuestro caso, definiremos la relevancia en base a la puntuación otorgada por el usuario: **consideramos relevante un artista que ha sido puntuado con al menos tres estrellas**.

Si un usuario en una misma sesión (es decir, en el mismo contexto) decide obtener otra lista de recomendaciones, no obtendría una lista idéntica a la anterior porque, cada vez que se genera una lista, se emplea una ponderación distinta del algoritmo de recomendación.

El parámetro que pondera las puntuaciones (o scores) de la recomendación es λ . Este parámetro combina las puntuaciones obtenidas por la recomendación basada en contenido y la recomendación basada en contexto. En esta evaluación, los valores de λ utilizados son:

- **$\lambda = 0$** : puntuación basada en contexto únicamente.
- **$\lambda = 0.25$** : el 25% de la puntuación es basada en contenido y el 75% es basada en contexto.
- **$\lambda = 0.5$** : el 50% de la puntuación es basada en contenido y el otro 50% es basada en contexto.
- **$\lambda = 0.75$** : el 75% de la puntuación es basada en contenido y el 25% restante es basada en contexto.
- **$\lambda = 1$** : puntuación basada únicamente en contenido.

Gracias a este factor, el mismo usuario en el mismo contexto puede obtener listas de recomendación completamente distintas.

En base a estas métricas, podremos comparar los distintos algoritmos, es decir, los distintos resultados que se obtienen con la ponderación de puntuaciones gracias al parámetro λ .

5.4.3 Conjunto de datos

Para comprender los resultados del experimento es importante definir previamente el conjunto de datos utilizado.

Usuarios

Este experimento se ha realizado con un total de **15 usuarios** con edades comprendidas entre los **8 y los 70 años**. Estos usuarios han puntuado a un total de 89 artistas en diversos contextos.

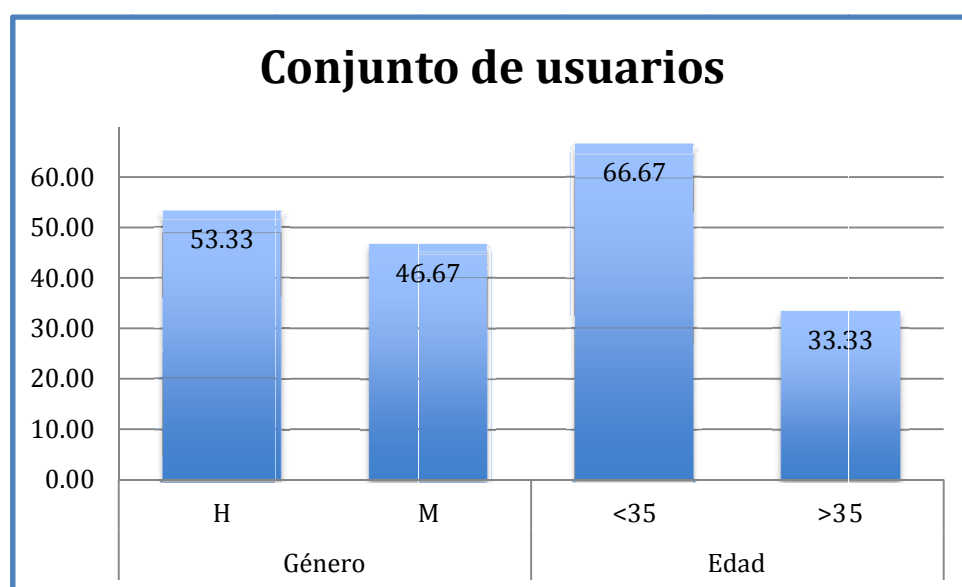


Ilustración 21 : Conjunto de usuarios

En esta tabla se muestran los porcentajes de los conjuntos donde se engloban los usuarios. En concreto, se han hecho particiones por género (hombre o mujer) y por edad (mayor o menor de 35 años).

Artistas

Los artistas empleados en este experimento se han obtenido gracias al proyecto de EC-Web ya mencionado. Estos artistas provienen de la base de datos de la página web de LastFM⁶. Los artistas que se incluyeron en el proyecto fueron los más populares de la web, es decir, los que tenían un mayor número de escuchas.

⁶ LastFM, <http://lastfm.es> - 5/VII/2014

Algunos de los artistas que se han incluido en la base de datos son:

ARTISTA	Género	Ponderación
Avenged Sevenfold	metal	0.39
	hardcore	0.33
	rock	0.28
Bruno Mars	soul	0.16
	pop	0.39
	rnb	0.45
Franz Schubert	jazz	0.09
	classical	0.91

Tabla 13 : Ejemplo de artistas en la base de datos

Se puede observar que no todos los artistas tienen asociados el mismo número de géneros: Schubert únicamente tiene dos (jazz y clásica), mientras que Bruno Mars y Avenged Sevenfold tienen tres. Todas las ponderaciones de los géneros de los artistas suman 1, es decir, si un artista tiene un único género, este tendrá una puntuación de 1.

Géneros musicales

Los **26 géneros musicales** empleados (ya descritos previamente en este documento) se distribuyen de la siguiente manera entre usuarios y artistas.

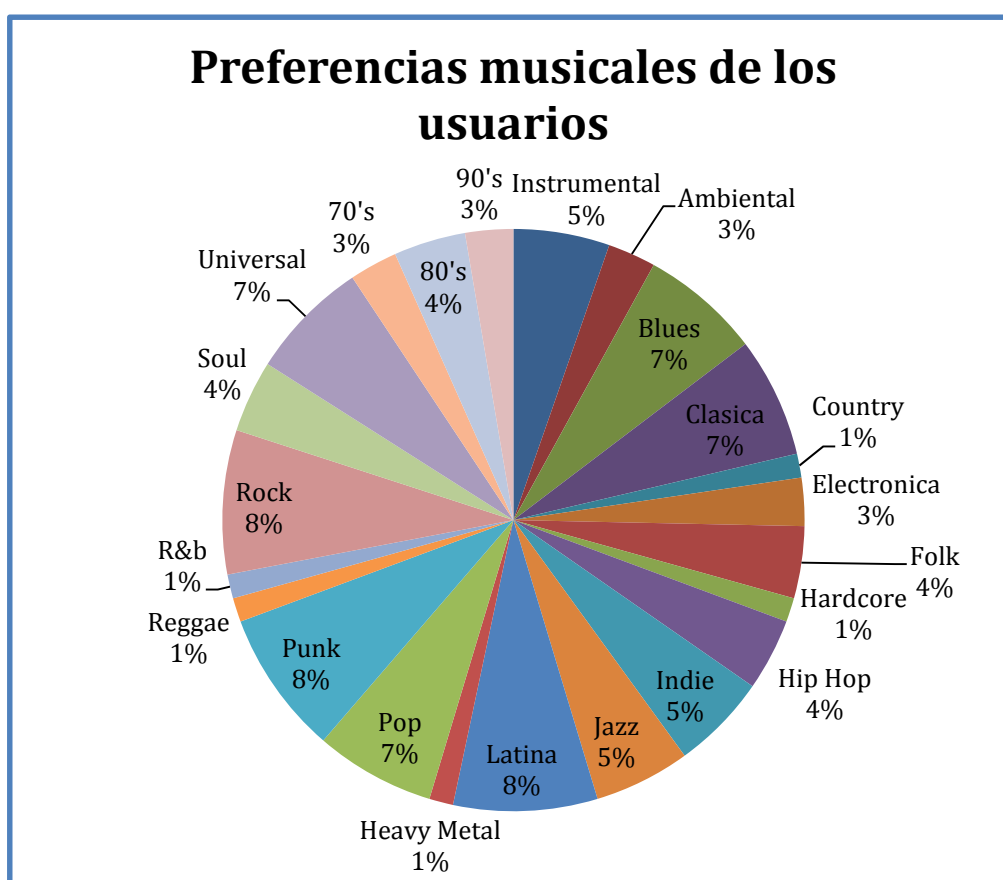


Ilustración 22 : Preferencias musicales de los usuarios

En este gráfico podemos observar la repartición de las preferencias musicales de los usuarios. El porcentaje muestra el número de usuarios que ha seleccionado ese género entre sus favoritos.

Los géneros más populares entre el grupo de usuarios donde se ha realizado el experimento son la música latina, punk y rock con un 8% y la pop, clásica, blues y universal con un 7%.

En el siguiente gráfico se muestran los géneros musicales de los 543 artistas empleados en el conjunto de datos utilizado en esta evaluación.

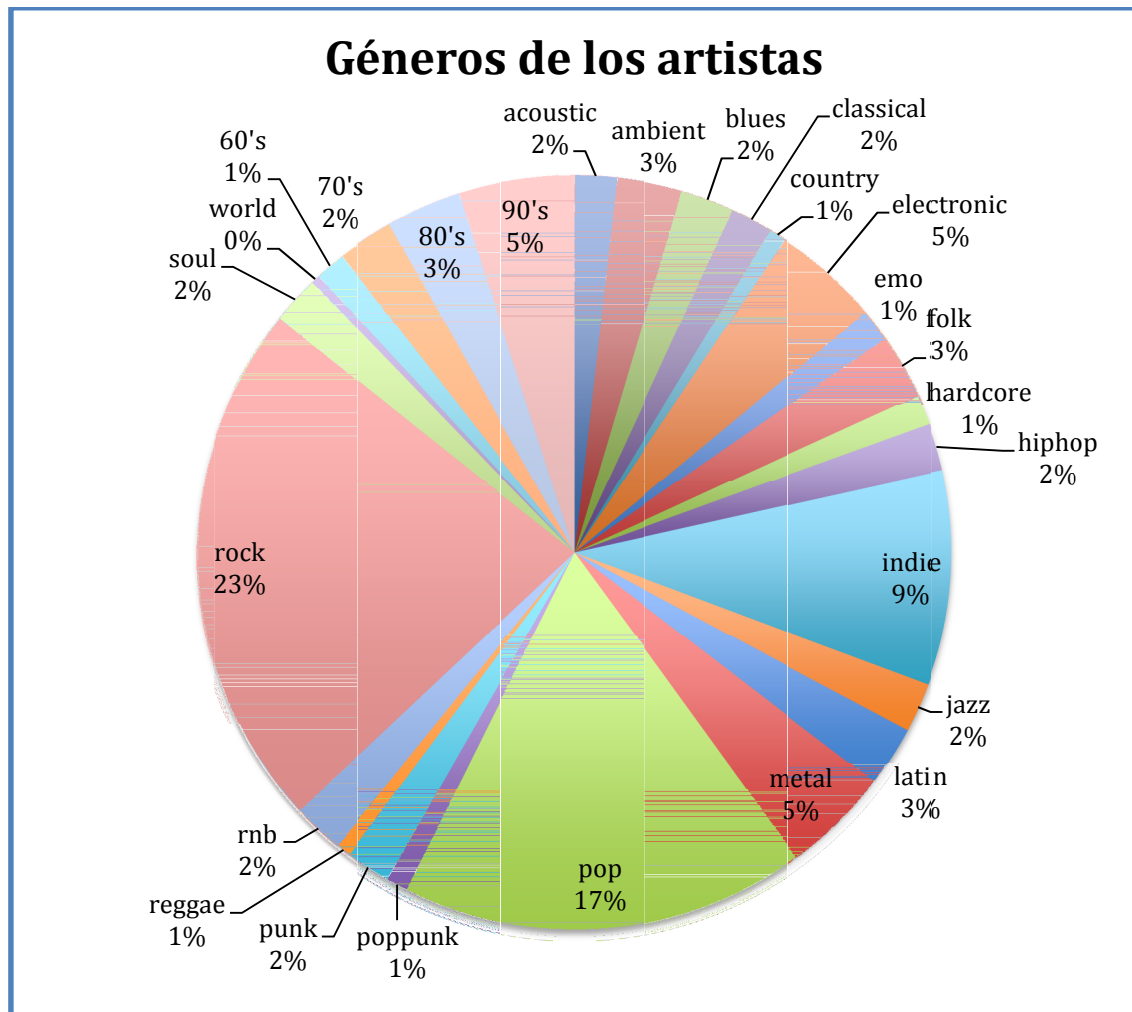


Ilustración 23 : Géneros musicales de los artistas

Como se puede observar, el género más popular entre los artistas es el rock, interpretado por un 23% de los autores, seguido por el pop, interpretado por un 17%. También se ve que el género menos tocado es el universal, la cifra ronda el 0%.

Contextos

Los contextos en este experimento se han definido por sus tres componentes: **contexto temporal**, **ubicación actual** del usuario y **estado de ánimo** actual del usuario.

En la siguiente tabla se muestran los datos del contexto temporal en porcentaje.

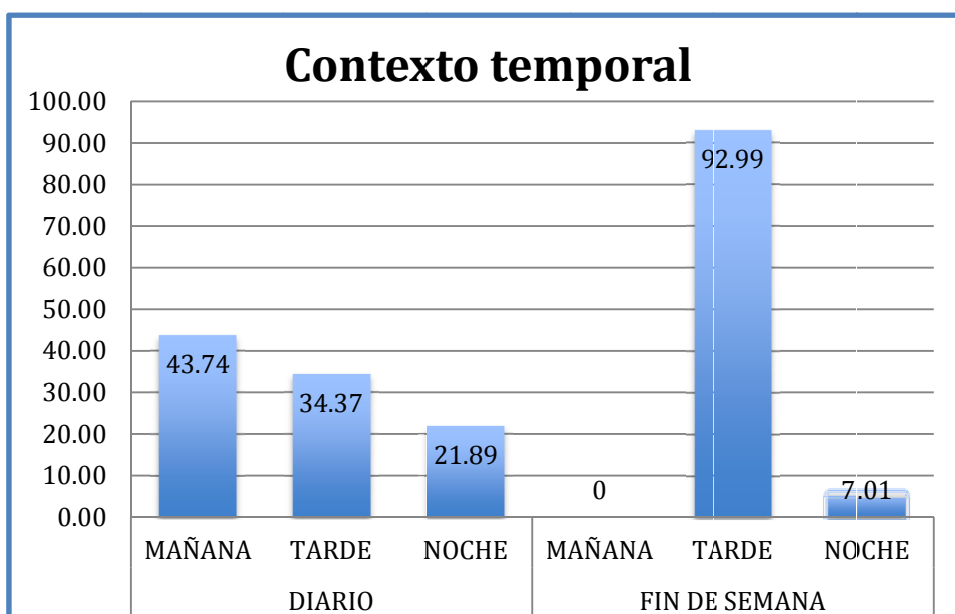


Ilustración 24 : Contexto temporal de los usuarios

Se puede observar que el contexto temporal donde se han realizado mayor número de evaluaciones han sido días de fin de semana por la tarde. No ha habido, sin embargo, ninguna evaluación una mañana de fin de semana. Los porcentajes en el período del día entre semana están más nivelados.

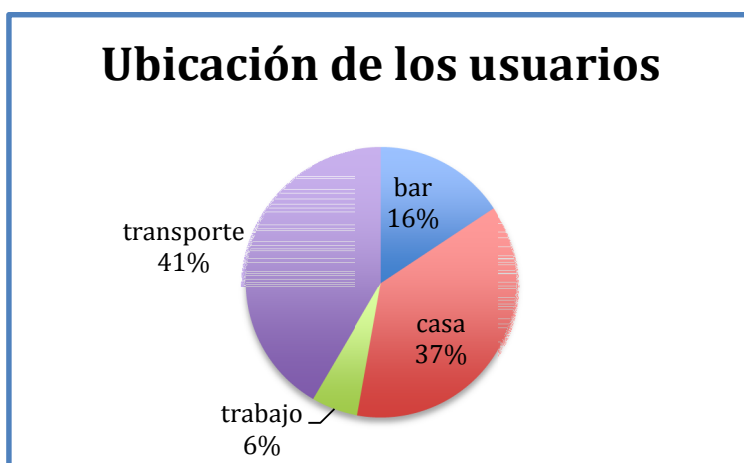


Ilustración 25 : Ubicación de los usuarios

En cuanto a la ubicación de los usuarios, se puede observar en este gráfico que las dos ubicaciones preferidas de los usuarios son transporte y casa.

Consideramos como “transporte” cualquier tipo de vehículo, en nuestro caso, los usuarios que marcaron “transporte” se encontraban en coches, buses y metros. La opción “bar”, se corresponde con cualquier tipo de bar, discoteca o restaurante.

La definición del estado de ánimo de un usuario es la parte más compleja del contexto, porque aunque se haya dado un número de estados de ánimo finitos, no siempre se ajustan al estado de ánimo actual del usuario. Otras veces, los usuarios están confusos sobre sus propios sentimientos y vacilan a la hora de anotar su estado de ánimo.

En la siguiente imagen se muestran los estados de ánimos iniciales de los usuarios.

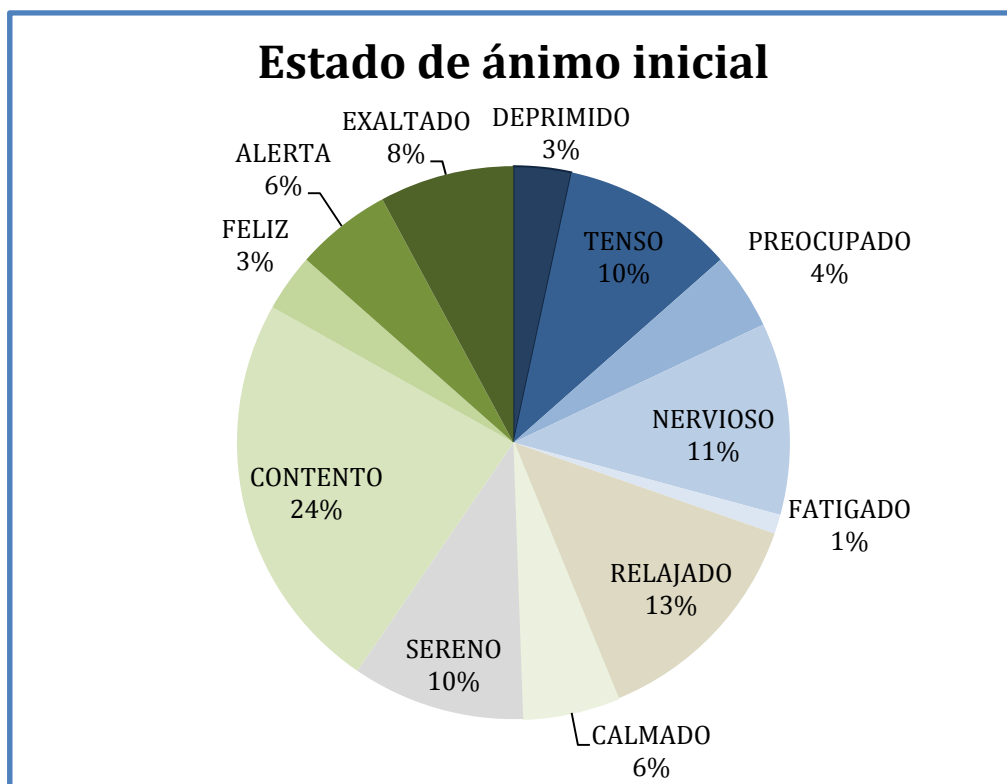


Ilustración 26 : Estado de ánimo inicial de los usuarios

En este gráfico podemos observar como el 24% de los usuarios comenzaban la recomendación de música en un estado de ánimo “contento”. Otros estados populares son relajado, nervioso, tenso y sereno.

Los estados con mayor excitación se han representado con colores más fuertes, mientras que los que tienen una excitación menor se han incluido en el diagrama con colores más suaves. Por otro lado, los valores con menor valencia (los considerados negativos) se han coloreado de distintos tonos de azul, mientras que los estados de ánimo que tienen una valencia positiva se han pintado de diferentes tonos de verde. Los colores con una valencia aproximada a cero, se han pintado en colores grises.

5.4.4 Resultados obtenidos

La primera cuestión que queríamos aclarar con esta evaluación era si **las recomendaciones basadas en contexto son mejores que las recomendaciones basadas en contenido**.

Para aclarar este punto, hemos observado las puntuaciones que los usuarios han dado a los artistas, concretamente las puntuaciones iguales o mayores que tres estrellas (nuestra definición de ítem relevante). En la siguiente figura se muestra el total de ítems considerados relevantes y no relevantes por algoritmo.

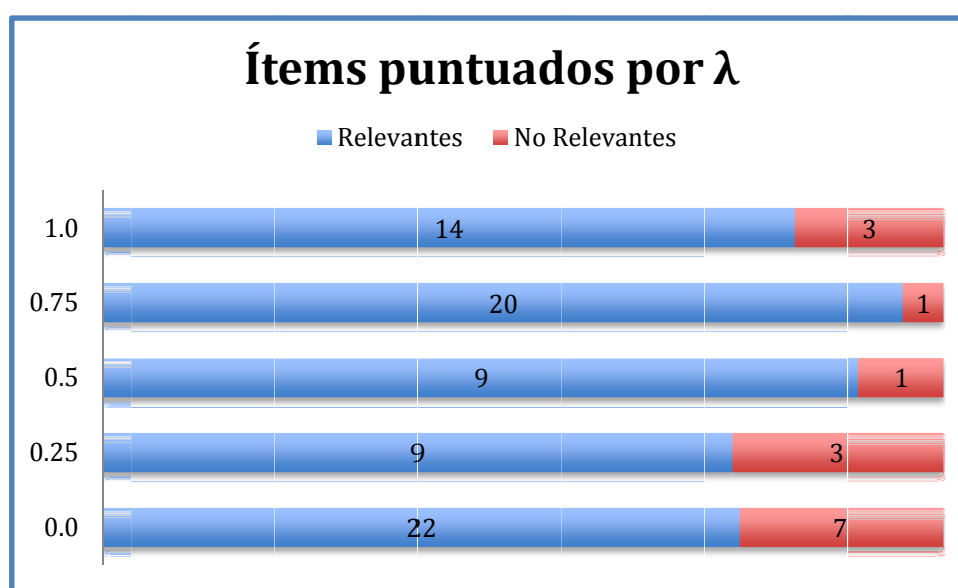


Ilustración 27 : Ítems puntuados por λ

Se puede ver en la figura que el número de ítems relevantes (74) es mucho mayor que el número de ítems no relevantes (15). Esto se debe a que la mayoría de los usuarios únicamente escuchaban artistas que ya les gustaban antes. Como en ningún momento se ha forzado a ningún usuario a puntuar a determinados artistas, se dan estos resultados.

Al ser un experimento limitado, el número de usuarios y de ratings no es muy elevado, es normal que se den estas descompensaciones entre las puntuaciones. Por este motivo, sería conveniente realizar otro experimento con un mayor número de usuarios para poder alcanzar unos resultados mejorados.

Gracias a estos valores hemos podido obtener las métricas descritas en el apartado de metodología para este conjunto de datos.

Precisión

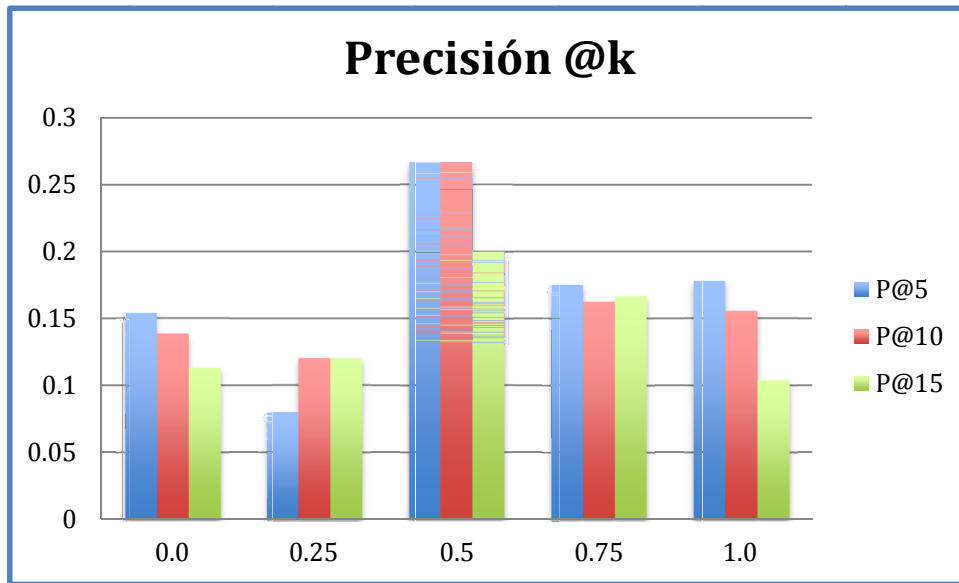


Ilustración 28 : Precisión @k

Como es de esperar, la precisión empeora cuanto más alta es la **k**. Esto se debe a la fórmula de la precisión: a mayor número de ítems devueltos, mayor es el denominador en la fórmula y, por lo tanto menor es el resultado.

Los mejores resultados de precisión se obtienen con $\lambda = 0.5$. Esto significa que la ponderación formada al 50% por las recomendaciones sensibles al contexto y las basadas únicamente en contenido es la más precisa.

Recall

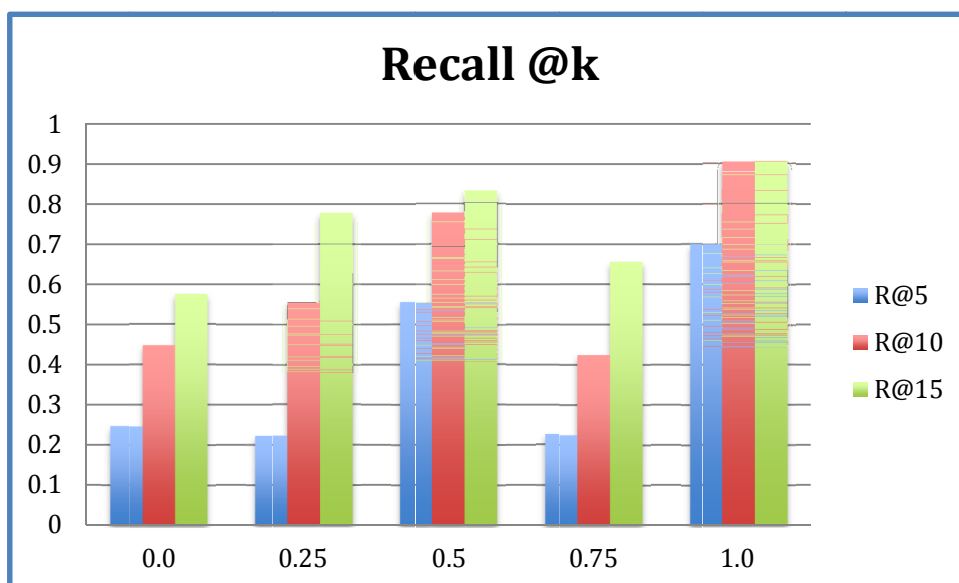


Ilustración 29 : Recall @k

Al contrario que en la precisión, cuanto mayor es la k mayor es el recall. Esto se debe a que cuanto más se amplíe el rango donde buscar ítems relevantes, más ítems relevantes se encontrarán.

El mejor valor de recall para este experimento se da en $\lambda = 1$. Esto es normal, ya que cuando se tienen en cuenta exclusivamente las preferencias del usuario se suele acertar más en las recomendaciones.

El valor de recall para $\lambda = 0$ es menor porque cuando se recomienda en base al contexto, se suelen recomendar ítems ya puntuados por otros usuarios y es posible que estos no tengan los mismos gustos musicales que el usuario actual.

Media armónica

Gracias a la media armónica se suavizan los valores de precisión y recall y se obtienen los verdaderos resultados.

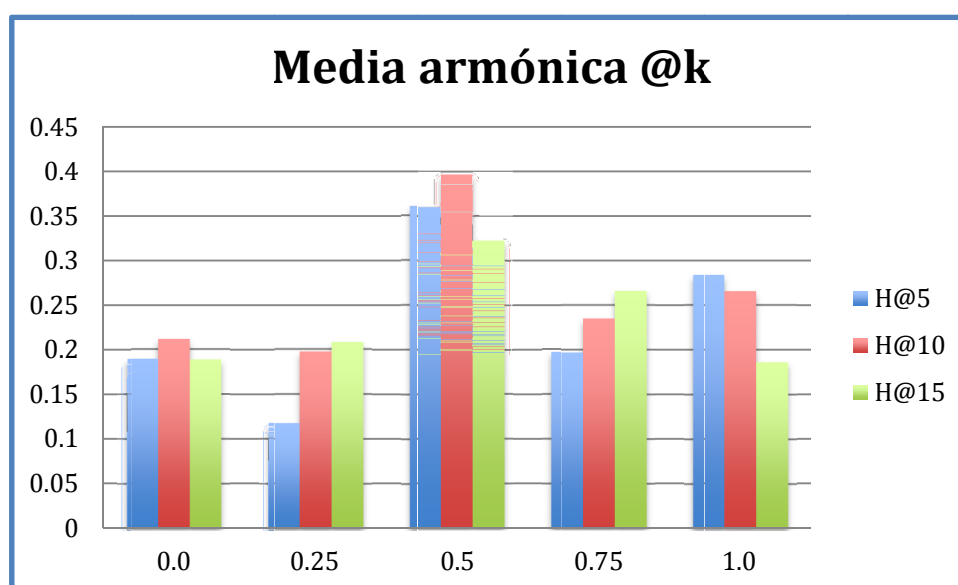


Ilustración 30 : Media armónica @k

Como se puede observar, hay una gran diferencia entre el algoritmo de $\lambda = 0.5$ y el resto de algoritmos. Con estos resultados, se puede concluir que no es mejor ni peor recomendar basándose en contexto, el resultado más eficiente se obtiene cuando se combinan ambas métricas. En este caso, podemos observar que los mejores resultados se obtienen combinándolas al 50%.

Otra pregunta que queríamos contestar es si **existe un patrón de transiciones de estados de ánimo**, es decir, si hay más probabilidad, estando en un estado, de quedarse en él o variar a otro.

Para contestar a este interrogante, se ha obtenido una tabla donde se muestran las transiciones entre estados de ánimos iniciales y finales de los usuarios.

	DEPRIMIDO	TRISTE	ESTRESADO	TENSO	PREOCUPADO	NERVIOSO	ADORMECIDO	FATIGADO	RELAJADO	CALMADO	SERENO	CONTENTO	FELIZ	ALERTA	EXCITADO	EXALTADO
DEPRIMIDO	1						1				1					
TRISTE																
ESTRESADO																
TENSO			1	4	1		1					1			1	
PREOCUPADO							1			1		2				
NERVIOSO						2			1	1	1		4	1		
ADORMECIDO																
FATIGADO								1								
RELAJADO	2		1						4			1	1		1	2
CALMADO		1					1			1		1			1	
SERENO		1									5	2	1			
CONTENTO			1			1			1	1		13	1			3
FELIZ													3			
ALERTA									1	1				2	1	
EXCITADO																
EXALTADO	1						1					1	2			2

Tabla 14 : Transiciones de estados de ánimos

Los estados de ánimo se han ordenado en base a su **valencia**: los primeros son los que tienen una valencia más baja (los considerados estados de ánimo negativos) y los últimos en la tabla son los que tienen una valencia mayor (estados de ánimo positivos).

Al tener pocos datos, es difícil obtener resultados de patrones de transiciones concretas. Es decir, resulta complicado concluir que el 90% de los usuarios que empiezan adormecidos cambian su estado de ánimo a preocupados. Sin embargo, sí que se pueden obtener conclusiones agrupando los estados de ánimo por valencia.

Como se puede observar en la tabla, la mayoría de las transiciones de estados de ánimo se encuentra en el cuadrante inferior derecho. En este cuadrante se agrupan los estados de ánimo positivos. También se puede comprobar que, dentro de los cuadrantes superiores, el derecho está ligeramente más poblado que el izquierdo.

En la siguiente imagen se muestra una representación gráfica de la tabla de transiciones de estado de ánimo agrupadas por valencia. Gracias a esta gráfica será más sencillo obtener las conclusiones.

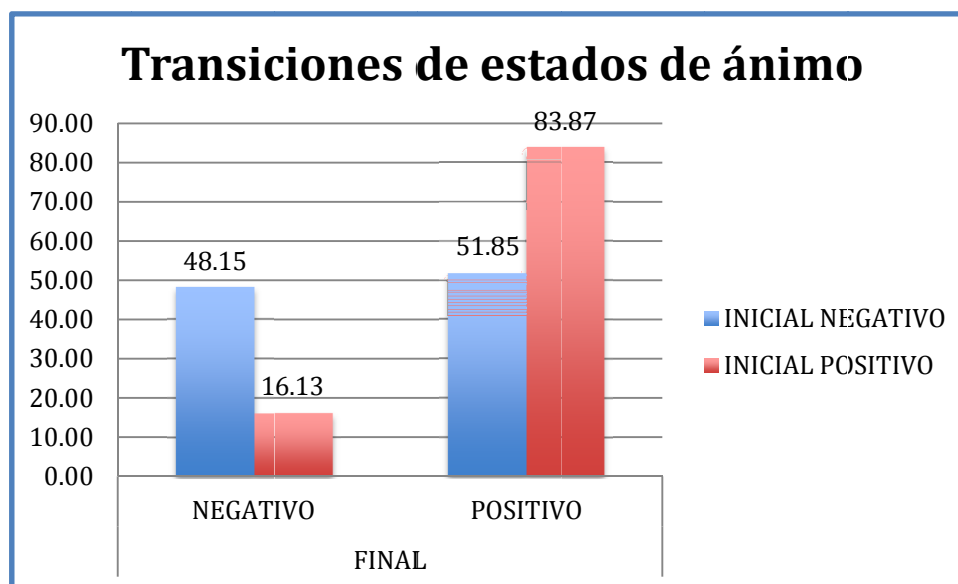


Ilustración 31 : Transiciones de estados de ánimo

En este gráfico se puede observar que el 83,87% de los usuarios que comienzan en un estado de ánimo positivo terminan con un sentimiento también positivo, mientras que tan solo el 16,13% de estos usuarios cambia la valencia de su estado de ánimo, es decir, pasan de un estado positivo a uno negativo.

En el lado contrario, prácticamente la mitad de los usuarios que comienzan con un estado de ánimo negativo se mantienen en este tipo de estado. La diferencia entre comenzar en un estado inicial negativo y pasar a uno positivo o negativo es prácticamente nula.

Analizando estos datos se puede concluir que los usuarios que comienzan con estado de ánimo positivo suelen permanecer en este y que, por el contrario, no existe un patrón de transición claro para los usuarios que comienzan con un estado de ánimo negativo.

Como curiosidad, se puede destacar el elevado número de usuarios que han comenzado en el estado de ánimo **contento** y han terminado en ese mismo estado. El 20% de los usuarios que han comenzado en un estado inicial **positivo**, lo han hecho con el sentimiento “contento”, y han terminado con este mismo sentimiento.

5.4.5 Discusión

Antes de comenzar la discusión de este experimento, es importante recordar la limitación de los datos con los que se ha realizado. En este caso, la evaluación ha sido ejecutada con un número reducido de usuarios a los que no se les ha pedido un número mínimo de evaluaciones (algunos han puntuado a 10 artistas y otros solo a uno). Por este motivo, se plantea realizar un experimento más exhaustivo para confirmar los resultados preliminares obtenidos con esta evaluación.

Gracias a este experimento se ha deducido que la mejor opción para realizar recomendaciones sensibles al contexto es ponderarlas con las preferencias del usuario. Se ha comprobado que la combinación de los dos algoritmos mejora ambas partes, y que, gracias a esto, se obtiene una recomendación personalizada muy favorable al usuario. En este caso, se ha obtenido que el valor óptimo de ponderación es el 50%.

También se ha aclarado si existen o no patrones de transiciones entre estados de ánimo. En este caso, por los pocos datos con los que cuenta el experimento, no se pueden obtener patrones de un estado de ánimo a otro, pero sí elaborar patrones basados en la valencia de dichos estados.

Cuando un usuario comienza en un estado de ánimo positivo, lo más común es que continúe en un estado con este tipo de valencia. Sin embargo, no se puede deducir un patrón para los usuarios que comienzan con un estado de ánimo negativo.

Conclusiones y trabajo futuro

En este apartado se precisan las conclusiones obtenidas a raíz del desarrollo del proyecto y las posibles modificaciones del mismo para su mejora.

6.1 Conclusiones

El objetivo de este trabajo es el diseño, implementación y evaluación de una aplicación móvil de recomendación contextualizada de música. En esta aplicación se puede crear un perfil de usuario, definiendo los gustos musicales del propio usuario, y en base al contexto del usuario actual, se recomienda una lista de artistas. Posteriormente, se permite al usuario ver vídeos de los artistas recomendados, ver la información de los artistas y puntuar dichos artistas.

El resultado de este trabajo es una aplicación móvil para dispositivos Android que cumple con la lista de requisitos descritos en el párrafo anterior. Gracias a los 15 usuarios y los datos de 543 artistas se ha podido efectuar una pequeña evaluación de los algoritmos implementados para averiguar si las recomendaciones sensibles al contexto son mejores que las que se basan únicamente en contenido y si existe algún patrón entre transiciones de estados de ánimo.

Después de realizar el experimento, podemos concluir que las recomendaciones sensibles al contexto y las basadas en contenido son muy similares, pero que se observa una notable mejoría si se combinan ambas. Por otro lado, también se puede concluir que no se ha encontrado un patrón de cambios de estado de ánimo individuales, pero sí se ha logrado obtener un patrón de transiciones colectivas. Es decir, agrupando los estados de ánimo según su valencia (positiva o negativa) se obtienen patrones de cambios. Concretamente, en los usuarios que comienzan con un estado de ánimo positivo, se ha observado la tendencia a permanecer en estados de ánimo con la misma valencia.

6.2 Trabajo futuro

Este proyecto es característico por sus múltiples aplicaciones y mejoras. Es posible ir añadiendo más módulos a la aplicación, tanto para mejorar sus funcionalidades actuales como para añadir nuevas.

En esta sección se comentarán algunas posibles modificaciones del proyecto para su mejora.

6.2.1 Modificación del sensor de localización

En la actualidad, el **sensor de localización** solo lee la selección que ha realizado el usuario mediante la pantalla de selección de ubicación (constituida por una selección implementada en RadioGroup, para que el usuario únicamente pudiera elegir una ubicación), sin embargo, sería mucho mejor inferir esta información.

Una manera de inferir la ubicación del usuario sería añadiendo una tabla a la base de datos con ubicaciones frecuentes del usuario, por ejemplo: trabajo, casa, gimnasio, etc. para así no tener que preguntarle al usuario dónde se encuentra.

Gracias a los teléfonos móviles, es bastante sencillo averiguar las coordenadas del dispositivo, y por tanto, las del usuario. La librería “Google play services” de Google ha hecho posible implementar un nuevo módulo de localización. Gracias a este módulo se obtienen las coordenadas del usuario (**geolocalización**) y se guardan en la base de datos.

Si las coordenadas actuales del usuario no se corresponden con una ubicación previamente insertada en la base de datos, se le preguntará dónde se encuentra. De esta manera, se sabrá cuándo está en casa, en el trabajo, en un parque o en el gimnasio.

También se implementará un módulo de “compañía”. La compañía influye en el estado de ánimo de las personas [3], y, por tanto, también influirá en la música que se debe recomendar.

Se guardará en base de datos la relación entre la compañía y la ubicación. Por ejemplo, si el usuario suele ir solo al gimnasio o si siempre va acompañado de amigos al parque.

Algunas posibles opciones de compañía son “con amigos”, “con mi pareja”, “con niños”, “solo”...

6.2.2 Mejora del perfil de usuario

El **perfil de usuario** almacenado ahora mismo en la base de datos es escaso. Un nombre de usuario no da información acerca de la persona. Por eso es importante la mejora del perfil de usuario. Una mejora bastante sencilla es la ampliación del perfil de usuario.

El **género**, la **edad**, la **nacionalidad** y la **lengua materna** son algunos ejemplos de posibles campos de ampliación del perfil de usuario.

El sexo y la edad también influyen en el estado de ánimo de las personas. Las emociones negativas se manifiestan menos cuanto mayor es la edad de la persona. Esto sucede porque la experiencia emocional suele mejorar con la edad [3]. Por otra parte, el hecho de ser hombre o mujer influye en el estado de ánimo de una manera mucho más significativa de lo que en un principio se podría pensar.

Las evidencias confirman las diferencias entre hombres y mujeres cuando se habla de reacciones emocionales. Las mujeres muestran mayores expresiones emocionales. Suelen experimentarlas de una manera más intensa y las exteriorizan más frecuentemente (tanto positivas como negativas). En contraste con los hombres, las mujeres también muestran más comodidad al expresar sus emociones. Finalmente, las mujeres también suelen ser mejores que los hombres leyendo señales no verbales.

Una explicación para estas evidencias puede ser la manera en la que los hombres y las mujeres se han educado a lo largo de los años: los hombres deben ser valientes y duros, mientras que las mujeres deben ser cálidas y amables.

Otra posible explicación es que, las mujeres tienen una mayor habilidad para leer a los demás y exteriorizar sus emociones. Una tercera explicación sería que, las mujeres tienen una mayor necesidad de aprobación social, por lo que tienden a mostrar más las emociones positivas, como la felicidad [3].

Gracias a la ampliación del perfil de usuario, se pueden probar otros modelos más elaborados de recomendación, como por ejemplo, el **filtrado colaborativo**.

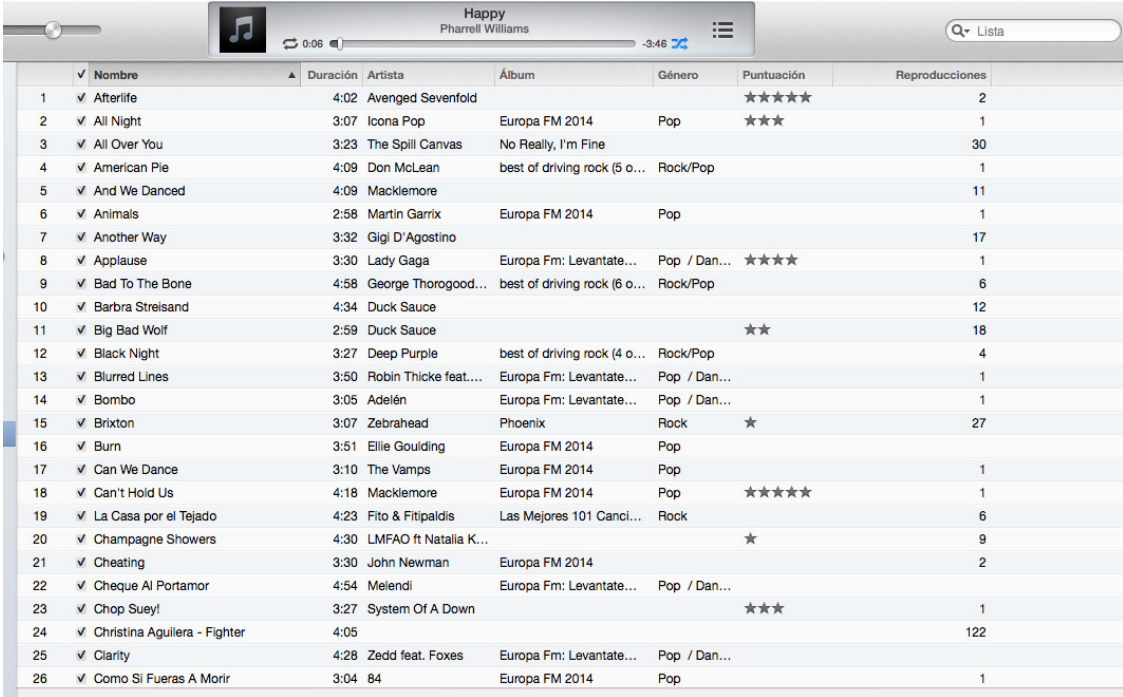
El filtrado colaborativo es un método de recomendación basado en la predicción de puntuaciones de un usuario para diversos ítems a través de la recopilación de las preferencias de otros usuarios.

El filtrado colaborativo se basa en la afirmación de que a los usuarios parecidos les gustan las cosas similares. Por ejemplo, si el usuario **x** tiene la misma opinión que el usuario **y** sobre el ítem **a**, es probable que tengan la misma opinión sobre otro ítem. En concreto, este tipo de técnica compara los usuarios para encontrar los más similares y así predecir las puntuaciones de ítems.

6.2.3 Inferencia de gustos musicales

Como ya se ha comentado previamente, una manera de mejorar la aplicación es **inferir los gustos musicales** del usuario a través de los metadatos.

Es posible obtener los géneros musicales de las canciones que se encuentran en el teléfono móvil del usuario y gracias a ellos, tener una idea de sus gustos. Otra posible opción es rastrear los metadatos de las páginas web con contenido musical que visite el usuario (YouTube, Lastfm, GrooveShark, Spotify...). Observando la música que escucha normalmente un usuario y dónde la escucha, se pueden obtener conclusiones rápidamente.



	✓ Nombre	Duración	Artista	Álbum	Género	Puntuación	Reproducciones
1	✓ Afterlife	4:02	Avenge Sevenfold			★★★★★	2
2	✓ All Night	3:07	Icona Pop	Europa FM 2014	Pop	★★★	1
3	✓ All Over You	3:23	The Spill Canvas	No Really, I'm Fine			30
4	✓ American Pie	4:09	Don McLean	best of driving rock (5 o...	Rock/Pop		1
5	✓ And We Danced	4:09	Macklemore				11
6	✓ Animals	2:58	Martin Garrix	Europa FM 2014	Pop		1
7	✓ Another Way	3:32	Gigi D'Agostino				17
8	✓ Applause	3:30	Lady Gaga	Europa Fm: Levantate...	Pop / Dan...	★★★★★	1
9	✓ Bad To The Bone	4:58	George Thorogood...	best of driving rock (6 o...	Rock/Pop		6
10	✓ Barbra Streisand	4:34	Duck Sauce				12
11	✓ Big Bad Wolf	2:59	Duck Sauce			★★	18
12	✓ Black Night	3:27	Deep Purple	best of driving rock (4 o...	Rock/Pop		4
13	✓ Blurred Lines	3:50	Robin Thicke feat....	Europa Fm: Levantate...	Pop / Dan...		1
14	✓ Bombo	3:05	Adelén	Europa Fm: Levantate...	Pop / Dan...		1
15	✓ Brixton	3:07	Zebrahead	Phoenix	Rock	★	27
16	✓ Burn	3:51	Ellie Goulding	Europa FM 2014	Pop		
17	✓ Can We Dance	3:10	The Vamps	Europa FM 2014	Pop		1
18	✓ Can't Hold Us	4:18	Macklemore	Europa FM 2014	Pop	★★★★★	1
19	✓ La Casa por el Tejado	4:23	Fito & Fitipaldis	Las Mejores 101 Canci...	Rock		6
20	✓ Champagne Showers	4:30	LMFAO ft Natalia K...			★	9
21	✓ Cheating	3:30	John Newman	Europa FM 2014			2
22	✓ Cheque Al Portamor	4:54	Melendi	Europa Fm: Levantate...	Pop / Dan...		
23	✓ Chop Suey!	3:27	System Of A Down			★★★	1
24	✓ Christina Aguilera - Fighter	4:05					122
25	✓ Clarity	4:28	Zedd feat. Foxes	Europa Fm: Levantate...	Pop / Dan...		
26	✓ Como Si Fuera A Morir	3:04	84	Europa FM 2014	Pop		1

Ilustración 32 : Ejemplo de metadatos musicales

Se puede deducir que a un usuario le gusta la música relajada cuando está en su casa, si siempre que se obtienen las coordenadas de su casa, el usuario escucha baladas y música ambiental.

Gracias a esta inferencia, se mejoraría el perfil del usuario, sus verdaderos gustos musicales, y, además, se podrían asociar gustos musicales a ubicaciones, lo cual influiría bastante en la recomendación, ya que no es lo mismo saber que a un usuario le gusta la música clásica y el rock, que saber que le gusta la música clásica cuando está en la biblioteca y la música rock cuando está en un bar.

6.2.4 Mejora del sensor de estados de ánimo

El **estado de ánimo** de una persona influye en todo lo que la rodea y también en cómo es la propia persona. Por ejemplo, la personalidad predispone a cada ser humano a experimentar ciertas emociones y estados de ánimo.

Realmente, cualquier persona puede sentir todo tipo de emoción y estado de ánimo, pero la personalidad hace que estemos predispuestos a experimentar ciertas emociones de manera más intensa [3].

Una posible mejora para el sensor de estados de ánimo sería someter al usuario a un pequeño test psicológico, sin embargo, los resultados podrían ser confusos y no siempre acertados. Además, no todos los usuarios estarían dispuestos a exponerse a un reconocimiento psicológico únicamente para escuchar música.

Por este motivo, la mejor manera de obtener el estado de ánimo de un usuario es a través de sus **publicaciones en redes sociales**.

Hoy en día, millones de usuarios en el mundo están registrados en redes sociales, dos de las más influyentes son Facebook (con 1110 millones de usuarios)⁷ y Twitter (con 500 millones de usuarios)⁸.

Se generan más de 340 millones de tweets al día. Esto supone una gran fuente de información de los usuarios. A todos estos datos se los conoce como **Big Data**. “Big Data es un término aplicado a conjuntos de datos que superan la capacidad del software habitual para ser capturados, gestionados y procesados en un tiempo razonable.”⁹

La captura, almacenamiento y filtrado de los datos procedentes del Big Data suelen ocasionar problemas. Pero, en nuestro caso no tendríamos que filtrar todos los datos: se filtrarían exclusivamente los datos publicados por el usuario actual.

Al restringir tanto el área de búsqueda de los datos, la captación se convierte en algo mucho más sencillo. Se le pediría permiso al usuario para el acceso a sus perfiles de las redes sociales y se evaluarían sus publicaciones recientes.

Una vez captados los datos, hay que filtrarlos para obtener el estado de ánimo. Esto se lleva a cabo mediante unos diccionarios de sentimientos. Tras el filtrado de los datos se puede inferir el sentimiento actual del usuario.

Una vez captado el estado de ánimo actual del usuario, se le recomendaría música para ese sentimiento concreto.

La captación de estados de ánimo en base a publicaciones en redes sociales puede sonar a algo vago, pero ya es una realidad.

⁷ Facebook, Wikipedia, <http://es.wikipedia.org/wiki/Facebook> - 5/VII/2014

⁸ Twitter, Wikipedia, <http://es.wikipedia.org/wiki/Twitter> - 5/VII/2014

⁹ Big Data, Wikipedia, http://es.wikipedia.org/wiki/Big_data - 5/VII/2014



Ilustración 33 : Lynguo

La aplicación **Lynguo** del Instituto de Ingeniería del Conocimiento ya hace tratamiento de datos para obtener diversos valores, entre ellos el sentimiento con el que un usuario ha publicado un comentario¹⁰.

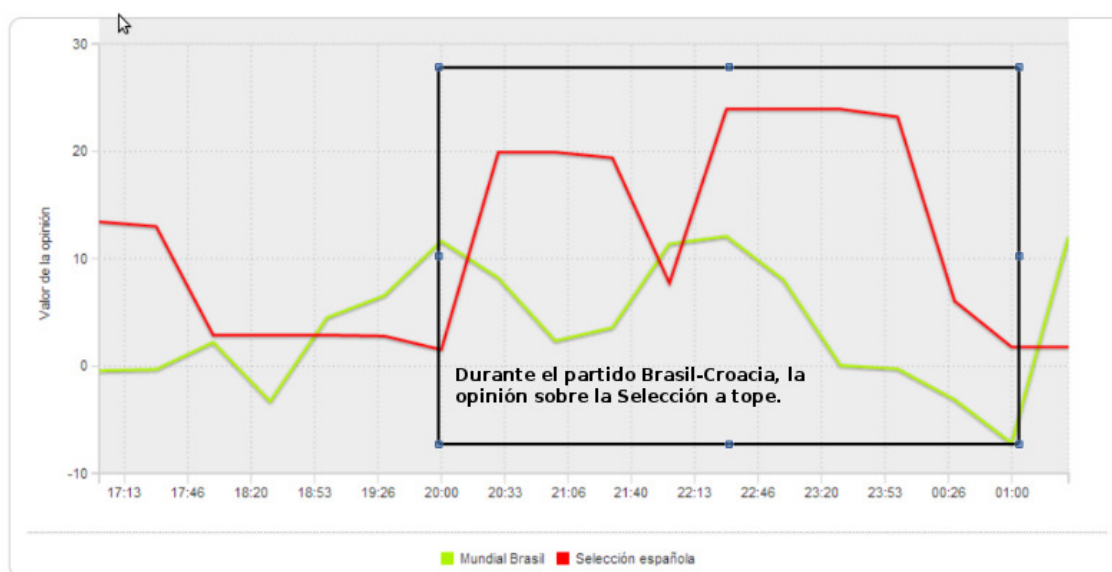


Ilustración 34 : Gráfica de Lynguo - Opinión de los usuarios sobre la Selección Española de fútbol durante el día 12 de junio de 2014

¹⁰ Lynguo, <http://www.iic.uam.es/es/soluciones-y-servicios/gestion-de-medios-sociales/lynguo> - 5/VII/2014



Ilustración 35 : Opinión de los usuarios sobre la proclamación del rey Felipe VI desde el 18 de junio hasta el 22 de junio de 2014

En el Instituto de Ingeniería del Conocimiento se han desarrollado diversos diccionarios para medir las opiniones de distintas marcas en redes sociales. En el caso de los sentimientos, un ejemplo de diccionario podría ser:

Expresión	DESPRECIO	ALIVIO	IMPRESIÓN	ASOMBRO
Mamarrachada	4			
Brotos verdes		3		
Flipar con				4
Piel de gallina			4	

Tabla 15 : Ejemplo de diccionario de sentimientos

En este ejemplo, que sería similar al que se implementaría en este trabajo, se hace una clasificación de expresiones relacionándolas con diversos sentimientos, y, dentro del sentimiento, se le da una ponderación.

Con este tipo de implementación es fácil obtener el sentimiento de una publicación de Facebook o de un tweet, lo que es más complicado es la construcción del diccionario.

Después de inferir el sentimiento, se podría preguntar al usuario si este se ajusta a su estado de ánimo actual. Gracias a esto, se afinaría mucho el análisis de sentimientos y por tanto, se recomendaría la música más ajustadamente al usuario.

6.2.5 Recomendación de canciones

En este trabajo se recomienda música en base a artistas musicales. Una mejora oportuna sería la **recomendación de canciones**.

Un mismo artista puede tocar más de un género. Es posible que haya canciones que combinen varios géneros, pero por lo general, cada canción tiene un único género musical. Además, las diferencias entre canciones del mismo artista pueden ser abismales.

Un ejemplo claro podría estar en el Heavy Metal. Este género es característico por el uso de la guitarra eléctrica. “La forma de tocar la guitarra es muy importante dentro del *heavy metal*. [...] El resultado, aunque simple, era en principio muy potente, lo que en definitiva constituía su objetivo.”¹¹

Sin embargo, estas canciones difieren mucho de las baladas de heavy metal. Las guitarras eléctricas suelen estar presentes, pero no son el centro de la canción y algunas veces son sustituidas por guitarras acústicas.

Por la gran variación que puede haber entre dos canciones de un mismo artista, es una gran mejora del sistema la recomendación de canciones, ya que, en definitiva, la recomendación es más precisa y fina.

6.2.6 Anotación de sentimientos para canciones y usuarios

Para poder obtener una información de la relación del usuario con la música sería muy adecuado dar la posibilidad de la anotación de canciones y artistas con sentimientos.

A través de estas anotaciones se podría saber qué sentimiento le evoca al usuario una canción o un artista.

Cada usuario es diferente, y por ello, la música que a un usuario le hace sentirse feliz, puede que a otro le haga aburrirse. Por este motivo es especialmente útil la anotación de sentimientos en música.

Por un lado, sabiendo que artistas y canciones evocan ciertos sentimientos a un usuario en concreto se puede modificar la recomendación. Se puede recomendar a un usuario música para los estados de ánimo más “favorables” o para tratar de influir en el estado de ánimo del usuario (pasar de triste a contento, por ejemplo).

Por otro lado, la información de las anotaciones de los usuarios es beneficiosa para crear una base de datos global. Si mil usuarios anotan el sentimiento “relajado” a una canción y tan solo diez la anotan como “nervioso”, se puede concluir que la canción es relajada. Esto puede servir para la recomendación a usuarios nuevos.

¹¹ Heavy Metal, Wikipedia, http://es.wikipedia.org/wiki/Heavy_metal - 5/VII/2014

Referencias

En este punto se detallan las referencias consultadas a la hora de la realización de este trabajo.

1. Adomavicius, G., & Tuzhilin, A.: Context-Aware Recommender Systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook*, pp. 217–253. Springer. (2011)
2. Adomavicius, G., & Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TKDE* 17(6), pp. 734–749 (2005).
3. Hume, D.: Emotions and Moods. In Robbins, S. P., Judge, T. A. (Eds.), *Organizational Behavior*, pp. 258–297.
4. Ignacio Fernández-Tobías, Iván Cantador, Laura Plaza: An Emotion Dimensional Model Based on Social Tags: Crossing Folksonomies and Enhancing Recommendations. 14th International Conference, EC-Web, pp. 88–100. (2013)
5. Koren, Y., & Bell, R. M.: Advances in Collaborative Filtering. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook*, pp. 145–186. Springer. (2011)
6. Lops, P., de Gemmis, M., & Semeraro, G.: Content-based Recommender Systems: State of the Art and Trends. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook*, pp. 73–105. Springer. (2011)
7. Ricci, F.: Mobile Recommender Systems. *International Journal of Information Technology and Tourism* 12(3), pp. 205–231. (2011)
8. Russell, J. A.: A Circumplex Model of Affect. *Journal of Personality and Social Psychology* 39(6), pp. 1161–1178. (1980)
9. Shani, G., Gunawardana, A.: Evaluating Recommendation Systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook*, pp. 257–297. Springer. (2011)
10. Song, Y., Dixon, S., Pearce, M.: A Survey of Music Recommendation Systems and Future Perspectives. 9th International Symposium on Computer Music Modelling and Retrieval (2012)